Variação de Parâmetros em Procedimentos GRASP

Marcelo Prais *Celso Carneiro Ribeiro*

Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro
Rio de Janeiro, RJ 22453-900, Brasil
prais@inf.puc-rio.br,celso@inf.puc-rio.br

Abstract
A greedy randomized adaptive search procedure (GRASP) is a metaheuristic for combinatorial optimization. In this paper we present an extensive computational study in which we explore the behavior of GRASP according to the variation of its main parameter. We discuss computational results obtained for classical instances of the traffic assignment problem in communication satellites, set covering, W-MAXSAT, and graph planarization. The results obtained show that the strategy of varying the value of the parameter along the iterations produces better solutions than considering a fixed value for all iterations.

Keywords: Otimização combinatória, GRASP, busca local, variação de parâmetros.

1 Introdução

Um procedimento de busca adaptativa gulosa e randomizada (GRASP) [8] é uma meta-heurística para problemas de otimização combinatória, composta de duas fases distintas. A primeira fase constrói uma solução viável, cuja vizinhança é explorada através de um processo de busca local. A melhor solução obtida ao longo de todas as iterações GRASP efetuadas é retornada como a solução encontrada.

Na fase de construção, uma solução viável é construída, elemento a elemento. A cada iteração deste processo de construção, os próximos elementos candidatos a serem incluídos na solução são colocados em uma lista, seguindo um critério de ordenação pré-determinado. O processo de seleção do próximo elemento da lista a ser incluído na solução que está sendo construída é baseado em uma função adaptativa gulosa, que estima o benefício da seleção de cada um dos elementos. A componente probabilística do procedimento reside no fato de que o elemento é selecionado de forma aleatória, a partir de um subconjunto restrito formado pelos ¨melhores¨

```
procedure GRASP(Tamanho_da_Lista,Max_Iterações,Semente_Aleatória)
1   Instância_de_Entrada();
2   for k = 1,..., Max_Iterações do
3       Constrói_Solução_Gulosa_Aleatória(Tamanho_da_Lista,Semente_Aleatória);
4       Busca_Local(Solução);
5       Atualiza_Solução(Solução,Melhor_Solução_Encontrada);
6   end for;
7   return Melhor_Solução_Encontrada;
end GRASP;
```

Figura 1: Pseudo-código do algoritmo GRASP básico

elementos que compõem a lista de candidatos (não necessariamente o melhor deles, que seria aquele selecionado por um algoritmo puramente guloso). Este subconjunto recebe o nome de *lista de candidatos restrita*.

Não se pode garantir a otimalidade local das soluções geradas pela fase de construção. Desta forma, quase sempre há um benefício em se aplicar uma busca local à solução construída. A busca local atua de forma iterativa, através da substituição sucessiva da solução corrente pela melhor em sua vizinhança. O processo de busca local termina quando não há mais soluções aprimorantes na vizinhança sendo visitada. A eficiência da busca local depende da qualidade da solução construída. O procedimento de construção tem então um papel importante na busca local, uma vez que as soluções construídas constituem bons pontos de partida para a busca local, permitindo assim acelerá-la.

O procedimento GRASP procura conjugar bons aspectos dos algoritmos puramente gulosos (isto é, convergência rápida na fase de busca local e soluções de boa qualidade), com aqueles de procedimentos aleatórios de construção de soluções (isto é, grande diversidade de soluções). A Figura 1 ilustra o procedimento GRASP básico, sob a forma de pseudo-código. O algoritmo tem como parâmetros de entrada o tamanho da lista de candidatos restrita, o número máximo de iterações e a semente utilizada pelo gerador de números pseudo-aleatórios. Após a leitura dos dados de entrada da instância (linha 1), as iterações GRASP são realizadas nas instruções contidas entre as linhas 2 e 6. Cada iteração consiste de uma fase de construção (linha 3), de uma fase de busca local (linha 4) e, se necessário, da atualização da melhor solução encontrada até o momento (linha 5). A melhor solução encontrada é retornada na linha 7.

Considere-se um problema de minimização sendo resolvido por um algoritmo guloso. Sejam c_{min} o elemento correspondente à seleção gulosa e c_{max} o elemento que implica no maior aumento de custo na função objetivo. A lista de candidatos restrita (LCR), a partir da qual um elemento será selecionado aleatoriamente, é definida em função de um parâmetro $0 \leq \alpha \leq 1$, compreendendo todos aqueles elementos que ainda não fazem parte da solução cujo impacto no valor da função objetivo esteja no intervalo $[c_{min}, \alpha(c_{max} - c_{min}) + c_{min}]$. Observe-se que $\alpha = 0$ implica em uma escolha puramente gulosa, já que a LCR conterá somente os elementos de custo c_{min}. Por outro lado, $\alpha = 1$ implica em uma escolha aleatória, já que a LCR conterá todos os elementos possíveis, definidos pelo intervalo $[c_{min}, c_{max}]$. Analogamente, no caso de um problema de maximização a LCR será definida pelo intervalo $[(1-\alpha)(c_{min} - c_{max}) + c_{max}, c_{max}]$, $0 \leq \alpha \leq 1$. Neste caso, $\alpha = 0$ implica em uma escolha aleatória a partir de uma LCR que contém todos os elementos possíveis, enquanto $\alpha = 1$ constitui-se em uma escolha puramente gulosa.

O parâmetro α que determina o tamanho da lista de candidatos restrita é basicamente o único parâmetro a ser ajustado na implementação de um procedimento GRASP. Feo e Resende [8] discutiram o efeito do valor de α na qualidade da solução e na diversidade das soluções geradas durante a fase de construção. Valores selecionados de α que levam a uma lista de candidatos restrita de tamanho muito limitado (ou seja, valor de α muito próximo da escolha gulosa) implicarão em soluções de qualidade muito próxima daquela puramente gulosa, com um esforço computacional proporcionalmente menor. Em contrapartida, provocam uma baixa diversidade de soluções construídas. Já uma escolha de α próxima da seleção aleatória leva a uma grande diversidade de soluções construídas. Por outro lado, muitas destas soluções terão qualidade inferior, tornando mais lento o processo de busca local.

Prais e Ribeiro [16] mostraram que o uso de um valor fixo para o parâmetro α freqüentemente impede a construção de soluções de melhor qualidade, que poderiam ser obtidas utilizando-se outros valores de α . Baseado nestes resultados, decidiu-se investigar o comportamento do procedimento GRASP em função de estratégias para variação do parâmetro α . Para isto, avalia-se o desempenho do GRASP para as seguintes estratégias de variação do parâmetro: (i) α auto-ajustado ao longo da execução do algoritmo; (ii) α selecionado aleatoriamente a cada iteração a partir de uma distribuição de probabilidades equiprovável; (iii) α selecionado aleatoriamente a cada iteração a partir de uma distribuição de probabilidades fixa, concentrada em torno do valor correspondente à escolha gulosa; e (iv) α fixo próximo da escolha gulosa. Resultados são obtidos e analisados para instâncias dos problemas de alocação de tráfego em sistemas de satélites TDMA, de recobrimento, de planarização de grafos e do problema MAXSAT com pesos.

Os problemas para os quais avaliou-se o desempenho de procedimentos GRASP em função das estratégias de variação de α são apresentados na Seção 2. As estratégias de variação de α são descritas na Seção 3. Na Seção 4, são apresentados os resultados computacionais dos experimentos efetuados. As conclusões do artigo estão na Seção 5.

2 Descrição dos problemas

Nesta seção, apresenta-se a descrição dos problemas para os quais estudou-se o comportamento do procedimento GRASP para diferentes estratégias de variação do parâmetro α. Os algoritmos para cada um destes problemas podem ser obtidos diretamente nas referências citadas.

2.1 Problema de decomposição de matrizes

O problema de decomposição de matrizes que surge no contexto da alocação de tráfego em sistemas de satélites baseados na tecnologia TDMA (multiplexação do sinal no domínio do tempo), aqui também chamado de problema TDMA, já foi estudado por diversos autores [1,2,6]. Seja uma matriz de tráfego $T = \{t_{ij}\}_{i=1,\dots,n}^{j=1,\dots,n}$ composta de m elementos não-nulos. Deseja-se encontrar uma decomposição de T na soma de q matrizes de modo de comutação, isto é:

$$T = \sum_{k=1}^{q} P^k,$$

satisfazendo às seguintes condições:

(i) para todo $k = 1, \dots, q$, P^k é uma matriz de modo de comutação, possuindo no máximo um elemento não-nulo por linha e coluna;

(ii) para todo $k = 1, \dots, q$, $p_{ij}^k > 0 \Rightarrow p_{ij}^k = t_{ij}$, isto é, não pode haver preempção e cada elemento não-nulo da matriz de tráfego original deve aparecer em uma e somente uma matriz de modo de comutação; e

(iii) a função objetivo $F(P^1, \dots, P^q) = \sum_{k=1}^{k=q} c_{max}(P^k)$ é minimizada, onde $c_{max}(P^k) = \max_{i=1,\dots,n; j=1,\dots,n}\{p_{ij}^k\}$, $k = 1, \dots, q$.

O número q de matrizes de modo de computação não é imposto, mas sim resultante do processo de decomposição. Rendl [17,20] demonstrou ser este problema NP-difícil. Este problema pode ser formulado como um problema de particionamento de grande porte [1,15,20]. Ribeiro et al. [20] usaram esta formulação para resolvê-lo de forma exata, através da combinação de um procedimento de geração de colunas com pesquisa arborescente. Resultados computacionais foram apresentados para até $n = 15$.

Prais e Ribeiro [16] propuseram recentemente um procedimento GRASP modificado para a resolução aproximada deste problema, chamado de GRASP Reativo, no qual o parâmetro α é auto-ajustado de acordo com a qualidade das soluções obtidas nas iterações precedentes. Sua fase de construção está baseada em uma heurística gulosa, aplicada diretamente sobre a matriz de tráfego T. Matrizes de modo de comutação são construídas sucessivamente até que todos os elementos originais de T tenham sido alocados. A alocação de um elemento de T a uma matriz de modo de comutação é feita escolhendo-se o maior elemento em T que ainda não foi selecionado e que não viole a restrição de cada matriz só poder ter um elemento não-nulo por linha e coluna. A aleatorização da fase de construção consiste em, ao invés de se

Problema	m	n	z^*	Problema	m	n	z^*
P.06	5	16	565	P.24	9	34	661
P.07	5	24	3771	P.25	9	36	504
P.08	5	24	4049	P.26	9	37	520
P.09	5	25	3388	P.27	9	44	216
P.10	6	16	3983	P.28	10	44	1729
P.11	6	18	3380	P.29	10	53	3470
P.12	6	26	657	P.30	10	60	4891
P.13	6	34	3220	P.31	11	47	620
P.14	7	31	3157	P.32	11	51	2480
P.15	7	34	341	P.33	11	56	3018
P.16	7	34	2343	P.34	12	74	1980
P.17	7	34	3281	P.35	12	86	2140
P.18	7	37	3228	P.36	12	101	7210
P.19	8	36	3710	P.37	13	87	6360
P.20	8	37	1830	P.38	13	94	2130
P.21	8	38	3660	P.40	14	85	4879
P.22	8	38	1912	P.41	14	116	2688
P.23	8	44	3770	P.42	15	138	2466

Tabela 2.1: Instâncias do problema de decomposição de matrizes

selecionar o maior elemento viável de T que ainda não foi alocado a uma matriz de modo de comutação, selecionar-se aleatoriamente um elemento da lista de candidatos restrita que contém os "maiores" elementos de T.

A fase de busca local é constituída de movimentos construtivos e de movimentos destrutivos. Os movimentos construtivos estão baseados em matrizes de modo de comutação que contêm estritamente menos do que n elementos não-nulos. Para cada uma delas, procura-se adicionar o maior número possível de elementos de T que não violem a restrição de somente um elemento não-nulo por linha e coluna. Uma nova solução é gerada aplicando-se o mesmo algoritmo de construção aos elementos originais de T que não foram selecionados. Movimentos destrutivos são aplicados a matrizes de modo de comutação com exatamente n elementos não-nulos. Para cada uma delas, selecionam-se um ou mais elementos para serem retirados e, a partir da matriz resultante, aplica-se o mesmo algoritmo de construção aos elementos originais de T que ainda não foram selecionados, completando-se assim a solução.

A Tabela 2.1 contém a descrição das instâncias que foram utilizadas nos estudos computacionais efetuados com o GRASP para o problema TDMA. Resultados computacionais são aqui relatados para 36 das 42 instâncias originalmente resolvidas de forma exata por Ribeiro et al. [20] (descartou-se instâncias muito pequenas para as quais $n < 5$, bem como a instância P.39, para a qual os dados não estavam disponíveis). Para cada problema, n representa a dimensão da matriz de tráfego, m

o número de elementos não-nulos da matriz de tráfego e z^* o valor da solução ótima encontrada pelo método híbrido de geração de colunas combinado com pesquisa arborescente.

2.2 Problema de recobrimento

Sejam n conjuntos finitos $P_1, P_2, \ldots, P_n$, $I = \cup(P_j : 1 \leq j \leq n) = \{1, \ldots, m\}$ e $J = \{1, \ldots, n\}$. Um subconjunto $J^* \subseteq J$ recobre J se $\cup(P_j : j \in J^*) = I$. O problema de recobrimento consiste então em encontrar um recobrimento de cardinalidade mínima. Em [10] mostra-se que a versão de decisão deste problema é NP-completo.

Fulkerson et al. [9] descreveram uma classe de problemas de recobrimento computacionalmente difíceis, que surge no cálculo da largura-1 de matrizes de incidência em sistemas de triplas de Steiner. A largura-β de uma matriz A de elementos 0-1 é definida como sendo o número mínimo de colunas que podem ser selecionadas de A, de modo que a soma dos elementos de cada uma das linhas da submatriz resultante seja pelo menos igual a β. As matrizes de incidência que surgem no contexto de sistemas de triplas de Steiner possuem exatamente três elementos iguais a um por coluna. Refere-se, então, a instâncias do problema de recobrimento no contexto dos sistemas de triplas de Steiner pelas suas matrizes de incidência.

Feo e Resende [7] apresentaram um procedimento GRASP para este problema, baseado na heurística gulosa proposta por Chvátal [3]. A partir dos n conjuntos finitos $P_1, P_2, \ldots, P_n$, a heurística gulosa de Chvátal seleciona o conjunto P_j não vazio de maior cardinalidade (aquele que cobre o maior número de elementos possíveis) e assim sucessivamente, até que todos os elementos de $P_1 \cup P_2 \cup \ldots \cup P_n$ tenham sido cobertos. A fase de construção do procedimento GRASP proposto por Feo e Resende em [7] difere da heurística de Chvátal quando da seleção do próximo conjunto P_j. Ao invés de selecionar-se aquele com cardinalidade máxima $card_{max}$, escolhe-se um conjunto selecionado aleatoriamente dentre aqueles que fazem parte de uma lista de candidatos restrita, composta de conjuntos cujas cardinalidades são no mínimo igual a $\alpha card_{max}$, $(0 \leq \alpha \leq 1)$. Resultados computacionais foram relatados para sete instâncias do problema de recobrimento: A_9, A_{15}, A_{27}, A_{45}, A_{81}, A_{135} e A_{243}. As quatro primeiras são provenientes de Fulkerson et al. [9]. As demais foram geradas utilizando o método recursivo de Hall [12]. Destes problemas, a solução ótima só é conhecida para os cinco primeiros. A Tabela 2.2 contém um resumo de suas características. Para cada problema A_i, m representa o número de restrições, n o de variáveis e $\underline{z}$ o valor da solução obtida com a heurística de Chvátal.

2.3 Problema MAXSAT com pesos

Sejam $C_1, C_2, \ldots, C_m$ m cláusulas envolvendo variáveis booleanas $x_1, x_2, \ldots, x_n$, que podem assumir somente valores *verdadeiro* ou *falso* (1 ou 0). A cada cláusula C_i está associado um peso não-negativo w_i. Definindo-se uma cláusula $C_i = \bigvee_{j=1}^{n_i} l_{ij}$ onde cada literal l_{ij} corresponde a uma das variáveis x_j ou ao seu complemento,

Problema	m	n	$\underline{z}$
A_9	9	12	5
A_{15}	15	35	9
A_{27}	27	116	19
A_{45}	45	330	31
A_{81}	81	1080	64
A_{135}	135	3015	107
A_{243}	243	9801	208

Tabela 2.2: Instâncias do problema de recobrimento

o problema de máxima satisfatibilidade com pesos (W-MAXSAT) consiste em determinar uma atribuição de valores *verdadeiro* ou *falso* às n variáveis, de forma a maximizar a soma dos pesos das cláusulas satisfeitas. Os problemas MAXSAT e W-MAXSAT são NP-difíceis [5,10].

Resende et al. [18] apresentam um GRASP para o problema MAXSAT com pesos. A fase de construção do GRASP é constituída de n iterações. A cada iteração, atribui-se a uma variável o valor 0 ou 1. A função adaptativa gulosa está baseada na idéia de maximizar o peso total das cláusulas ainda não satisfeitas e que se tornam satisfeitas com a atribuição de valor a uma variável. A lista de candidatos restrita do GRASP para o MAXSAT com pesos considera dois tipos de restrição: restrição pelo valor e restrição pela cardinalidade. A busca local está baseada em trocas. Considerando-se a solução corrente, verifica-se qual o ganho quando uma variável tem seu valor alterado. Cada vez que o ganho for positivo, obtém-se uma solução vizinha de melhor qualidade do que a corrente.

Os problemas-testes utilizados nos estudos computacionais para o problema MAXSAT com pesos são derivados da classe de instâncias SAT jnh do 2^{nd} DIMACS Implementation Challange [13] e possuem 100 variáveis e de 800 a 900 cláusulas. Os pesos das cláusulas são inteiros uniformemente distribuídos entre 1 e 1000. Os experimentos descritos em [18] foram limitados a um subconjunto do conjunto jnh de teste. Para as instâncias jnh1 e jnh4 a jnh9, o número de cláusulas é $m = 850$. Para as instâncias jnh201 a jnh203, jnh205 e jnh207 a jnh220, $m = 800$. Para as instâncias jnh301 a jnh310, $m = 900$.

2.4 Problema de planarização de grafos

Um grafo é planar se pode ser desenhado no plano sem que haja cruzamentos de arestas. Dado um grafo $G = (V, E)$, onde $|V|$ é o número de vértices e $|E|$ o número de arestas, o problema de planarização de grafos consiste em encontrar um subconjunto $F \subseteq E$ de menor cardinalidade possível, de modo tal que o grafo $G' = (V, E \setminus F)$ seja planar. O problema de planarização de grafos é NP-difícil [14] e várias heurísticas foram propostas para sua resolução. Resende e Ribeiro [19] propõem um procedimento GRASP, que constitui uma extensão da heurística de duas fases de Goldschmidt e Takvorian (aqui chamada de 2F-GT) [11], descrita de forma sucinta a seguir.

A primeira fase da heurística 2F-GT é baseada em um algoritmo guloso para produzir uma seqüência de vértices. Conforme observado por Goldschmidt e Takvorian, a seqüência de vértices influencia o tamanho do subgrafo planar encontrado. De forma a produzir outras seqüências, talvez melhores, a randomização proporcionada pelo GRASP na fase de construção pode ser aplicada à primeira fase da 2F-GT, conforme explorado por Resende e Ribeiro [19]. Desta forma, ao invés de selecionar-se sempre o vértice de grau mínimo d_{min} na seqüência de vértices, seleciona-se o próximo vértice da seqüência aleatoriamente da lista de candidatos restrita definida pelos vértices cujos graus estejam no intervalo $[d_{min}, \alpha(d_{max}-d_{min})+d_{min}]$, para $0 \leq \alpha \leq 1$.

Os problemas-testes utilizados nos estudos computacionais para o problema de planarização de grafos estão descritos em [4,11]. Do conjunto de 75 problemas relatados em [19], foram utilizados apenas 37, por restrições no tempo de processamento disponível. A Tabela 2.4 contém uma descrição das características dos grafos utilizados. Para cada instância, lista-se o nome do problema, o número de vértices $|V|$ e de arestas $|E|$ do grafo, o número de arestas $|E \setminus F|$ da melhor solução publicada na literatura e uma indicação de se esta melhor solução conhecida é ótima. Os grafos cimi-g1 a cimi-g6 são grafos especiais coletados por Cimikowsky [4]. As instâncias g2 a g12 são grafos hamiltonianos que Goldschmidt e Takvorian utilizaram para testar a heurística 2F-GT. As instâncias rg50.1 a rg50.5 e rg75.1 a rg75.5 são grafos não-planares gerados aleatoriamente por Cimikowsky. As instâncias tg100.1 a tg100.10 são grafos não-planares que possuem um subgrafo planar máximo de tamanho $3|V|-6$. Estas instâncias estão relatadas em [4].

3 Estratégias de variação do parâmetro α

As aplicações de GRASP relatadas na literatura utilizam sempre um único valor fixo para o parâmetro α durante a execução de todas as iterações do procedimento. Este valor fixo é simplesmente arbitrado (conforme Seção 1) ou resultante de algum processo de calibragem para as instâncias objeto do estudo, em geral próximo do valor correspondente à escolha puramente gulosa. O processo de calibragem requer em geral um grande esforço computacional.

Prais e Ribeiro [16] observaram que a execução do GRASP com α fixo e próximo da escolha puramente gulosa não gerava soluções suficientemente diversas para permitir que soluções ótimas fossem encontradas. Além disso, observaram que a execução do GRASP com outros valores de α (distintos do valor inicialmente arbitrado) permitia freqüentemente encontrar a solução ótima. Com base nesta observação, propuseram o procedimento GRASP Reativo, no qual o parâmetro α é auto-ajustado à medida que iterações vão sendo realizadas, em função da qualidade

Problema	$\|V\|$	$\|E\|$	$\|E \setminus F\|$	Ótimo?	Problema	$\|V\|$	$\|E\|$	$\|E \setminus F\|$	Ótimo?
cimi-g1	10	21	19	Sim	rg50.1	50	123	91	?
cimi-g2	60	166	165	Sim	rg50.2	50	145	94	Não
cimi-g3	28	75	73	Sim	rg50.3	50	157	100	Não
cimi-g4	10	22	20	Sim	rg50.4	50	171	102	Não
cimi-g5	45	85	82	Sim	rg50.5	50	183	105	?
cimi-g6	43	63	59	Sim	rg75.1	75	196	129	Não
g2	45	85	82	Sim	rg75.2	75	202	137	?
g3	10	24	24	Sim	rg75.3	75	215	135	?
g4	10	25	24	Sim	rg75.4	75	256	136	Não
g5	10	26	24	Sim	rg75.5	75	266	140	Não
g6	10	27	24	Sim	tg100.1	100	304	294	Sim
g7	10	34	24	Sim	tg100.2	100	314	294	Sim
g8	25	69	69	Sim	tg100.3	100	324	294	Sim
g9	25	70	69	Sim	tg100.4	100	334	294	Sim
g10	25	71	69	Sim	tg100.5	100	344	292	Não
g11	25	72	69	Sim	tg100.6	100	354	287	Não
g12	25	90	69	Sim	tg100.7	100	364	279	Não
					tg100.8	100	374	281	Não
					tg100.9	100	384	279	Não
					tg100.10	100	394	277	Não

Tabela 2.4: Instâncias do problema de planarização de grafos

das soluções obtidas nas iterações precedentes. Ao invés de utilizar um valor fixo para α, propõem que α seja selecionado aleatoriamente a partir de um conjunto discreto $\Psi = \{\alpha_1, \ldots, \alpha_m\}$ contendo m valores pré-determinados. A utilização de valores distintos de α em iterações diferentes permite a construção de listas de candidatos restritas diferentes, eventualmente permitindo a geração de soluções diferentes que não seriam alcançadas através da utilização de um único valor fixo. Por exemplo, considere-se $\alpha_1 = 0,1; \alpha_2 = 0,2; \ldots; \alpha_m = 1$. Seja p_i a probabilidade associada à escolha de $\alpha_i, i = 1, \ldots, m$. Considere-se ainda os valores iniciais de $p_i = 1/m$, $i = 1, \ldots, m$. Propõe-se atualizar periodicamente as probabilidades $p_i, i = 1, \ldots, m$, utilizando para isso as informações coletadas durante a busca. Dentre diversas alternativas para efetuar esta atualização, Prais e Ribeiro utilizam uma regra denominada de *qualificação absoluta*. Seja z^* o valor da melhor solução encontrada até o momento e A_i o valor médio das soluções obtidas utilizando-se $\alpha = \alpha_i$ na fase de construção. A distribuição de probabilidades é atualizada periodicamente a cada bloco de um certo número pré-fixado de iterações realizadas, através da expressão

$$p_i = q_i / (\sum_{j=1}^{m} q_j),$$

onde $q_i = z^*/A_i, i = 1, \ldots, m$.

Quanto mais adequado for um valor $\alpha = \alpha_i$ (isto é, quanto menor for o valor de A_i), maior será o valor de q_i e, conseqüentemente, maior será o valor recalculado para a probabilidade p_i. Desta forma, no próximo bloco de iterações a serem efetuadas, valores de α que levaram a melhores soluções têm maior probabilidade de serem selecionados e, por conseguinte, são utilizados com maior freqüência na fase de construção. Este processo recebe o nome de GRASP Reativo, já que o valor de α não é fixo, mas sim auto-ajustado em função da qualidade das soluções encontradas durante a busca.

Baseado nos resultados obtidos para o problema TDMA, decidiu-se investigar para outros problemas o comportamento do GRASP em função de estratégias para variação do parâmetro α. As seguintes estratégias foram consideradas:

- Reativa (R): α auto-ajustado ao longo da execução do algoritmo (GRASP Reativo)

- Equiprovável (E): α selecionado aleatoriamente a cada iteração a partir de uma distribuição de probabilidades equiprovável e inalterada

- Híbrida (H): α selecionado aleatoriamente a cada iteração a partir de uma distribuição de probabilidades fixa, concentrada em torno de valor próximo à escolha gulosa

- Fixa (F): α fixo próximo da escolha gulosa

A estratégia de utilização de um único valor fixo de α próximo da escolha gulosa é aquela de menor esforço computacional. Isto, porque as soluções construídas são em geral de boa qualidade, reduzindo o esforço da busca local. Porém, implica em uma baixa diversidade de soluções construídas, tendo em vista o tamanho relativamente reduzido da lista de candidatos restrita. A estratégia Reativa é a única que permite que o valor de α seja auto-ajustado ao longo da execução das iterações, em função da qualidade das soluções obtidas nas iterações precedentes, proporcionando maior aderência às especificidades de cada instância.

4 Resultados computacionais

São apresentados a seguir os resultados computacionais obtidos para as quatro classes de problemas estudados. Ao procedimento GRASP básico foram adicionadas as rotinas de reavaliação da distribuição de probabilidades. As instruções relativas à gerência do procedimento GRASP Reativo são também incorporadas ao programa principal, que seleciona um valor para α a cada iteração da fase de construção e, a cada bloco de 1000 iterações efetuadas, chama as rotinas de atualização da distribuição de probabilidades.

Para o problema de decomposição de matrizes, utilizaram-se os algoritmos e códigos descritos em [16]. O código Fortran para o problema de planarização de grafos corresponde à implementação dos procedimentos descritos em Ribeiro e Resende [19], adaptado para a avaliação de estratégias de variação do parâmetro α. Para os demais problemas aqui tratados, os códigos Fortran foram cedidos por M.G.C. Resende e adaptados para o estudo em tela.

Os experimentos computacionais foram conduzidos em um ambiente de *main-frame* IBM 9672, modelo R34. Os códigos estão escritos em Fortran 77 e foram compilados com o compilador IBM VS Fortran (versão 2, *release* 5). Utilizou-se em todos os casos a rotina Fortran de geração de números aleatórios de Schrage [21].

Em todos os experimentos, utilizou-se um número máximo de iterações igual a 10000 e a semente inicial para a geração dos números pseudo-aleatórios igual a 270001. No caso da estratégia Reativa, a distribuição de probabilidades inicial é equiprovável. Para cada problema e para cada uma das estratégias, são apresentados o valor z^* da melhor solução encontrada após 10000 iterações e o tempo de processamento em segundos (seg's). A penúltima coluna de cada uma das tabelas de resultados contém uma indicação de qual estratégia proporcionou o melhor resultado do GRASP e a última coluna indica se a melhor estratégia levou a um ótimo global ou não.

4.1 Problema de decomposição de matrizes

Para o problema de alocação de tráfego em sistemas de satélites TDMA, a estratégia denominada Fixa utiliza $\alpha = 0,1$ em todas as iterações da fase de construção. Para a estratégia Híbrida, o valor de α é selecionado aleatoriamente a partir da seguinte distribuição de probabilidades, concentrada em torno da escolha gulosa: $p(\alpha_1 = 0,1) = 0,5$; $p(\alpha_2 = 0,2) = 0,25$; $p(\alpha_3 = 0,3) = 0,125$; $p(\alpha_4 = 0,4) = 0,03$; $p(\alpha_5 = 0,5) = 0,03$; $p(\alpha_6 = 0,6) = 0,03$; $p(\alpha_7 = 0,7) = 0,01$; $p(\alpha_8 = 0,8) = 0,01$; $p(\alpha_9 = 0,9) = 0,01$; $p(\alpha_{10} = 1) = 0,005$.

A Tabela 4.1 contém os resultados para cada uma das instâncias apresentadas na Tabela 2.1. Pelos resultados apresentados na penúltima coluna, observa-se que a variação do valor do parâmetro α surte efeito, já que a utilização de um valor fixo de α nunca supera as soluções obtidas com a variação do parâmetro. As estratégias Reativa e Equiprovável foram aquelas que produziram os melhores resultados. Em particular, para as instâncias P.23 e P.38 a estratégia Equiprovável encontrou as melhores soluções, enquanto que a Reativa encontrou a melhor solução para a instância P.42. Em termos do tempo de processamento, como era esperado, o GRASP com α fixo levou o menor tempo de processamento (358,2 segundos em média para as 36 instâncias), sendo porém aquele que teve o pior desempenho. A estratégia Reativa levou em média 579,0 segundos (segundo menor tempo de processamento dentre as quatro estratégias), com desempenho praticamente equivalente ao da estratégia Equiprovável, com tempo médio de processamento inferior em cerca de 11,5% a este último (que levou em média 642,8 segundos). A estratégia Híbrida levou em média 612,6 segundos, com desempenho superior somente ao da estratégia Fixa, porém com o segundo menor tempo de processamento.

A última coluna da tabela indica que a estratégia de se variar o valor de α para o GRASP é eficaz, já que a solução ótima exata só não foi encontrada para as instâncias P.13, P.20, P.40 e P.41. A estratégia Fixa nunca obtém isoladamente a solução ótima.

4.2 Problema de recobrimento

Neste caso, a estratégia Fixa utiliza $\alpha = 0,9$. Para a estratégia Híbrida, o valor de α é selecionado aleatoriamente a partir da seguinte distribuição de probabilidades, concentrada em torno da escolha gulosa: $p(\alpha_1 = 0,1) = 0,005$; $p(\alpha_2 = 0,2) = 0,01$; $p(\alpha_3 = 0,3) = 0,01$; $p(\alpha_4 = 0,4) = 0,01$; $p(\alpha_5 = 0,5) = 0,03$; $p(\alpha_6 = 0,6) = 0,03$; $p(\alpha_7 = 0,7) = 0,03$; $p(\alpha_8 = 0,8) = 0,125$; $p(\alpha_9 = 0,9) = 0,25$; $p(\alpha_{10} = 1) = 0,5$.

Os resultados obtidos para as instâncias da Tabela 2.2 são apresentados na Tabela 4.2. A estratégia que proporcionou o melhor desempenho foi a Reativa (encontrou isoladamente a melhor solução para a instância A_{135}. Para a instância A_{243}, a estratégia Fixa não foi capaz de encontrar o menor recobrimento. Da mesma forma que para o problema de decomposição de matrizes, o GRASP conjugado à estratégia de variação de α produz melhores resultados do que a estratégia de se utilizar um valor fixo de α.

4.3 Problema MAXSAT com pesos

Para este problema, a estratégia Fixa utiliza $\alpha = 0,9$. Para a estratégia Híbrida, o valor de α é selecionado aleatoriamente a partir da seguinte distribuição de probabilidades, concentrada em torno da escolha gulosa: $p(\alpha_1 = 0,1) = 0,005$; $p(\alpha_2 = 0,2) = 0,01$; $p(\alpha_3 = 0,3) = 0,01$; $p(\alpha_4 = 0,4) = 0,01$; $p(\alpha_5 = 0,5) = 0,03$; $p(\alpha_6 = 0,6) = 0,03$; $p(\alpha_7 = 0,7) = 0,03$; $p(\alpha_8 = 0,8) = 0,125$; $p(\alpha_9 = 0,9) = 0,25$; $\alpha_{10} = 1) = 0,5$. Os resultados obtidos para as instâncias referidas na Seção 2.3 estão apresentados na Tabela 4.3, divididos em três classes de instâncias: jnh1 e jnh4 a jnh19; jnh201 a jnh203, jnh205 e jnh207 a jnh220; e jnh301 a jnh310.

A qualidade média das soluções obtidas para as três classes de instâncias jnh e para cada uma das quatro estratégias de variação do parâmetro α permite concluir que a variação de α também neste caso produz melhores resultados do que a utilização de um valor fixo e único. Uma comparação estrita entre as três estratégias de variação de α permite também concluir que, para este problema e para as instâncias utilizadas, a estratégia Reativa foi a que levou ao melhor desempenho médio do GRASP.

Em termos do tempo de processamento, o menor tempo médio corresponde ao da estratégia Fixa, seguido pela estratégia Híbrida. O esforço computacional requerido pela estratégias Reativa e Equiprovável é aproximadamente o mesmo e cerca de 50% superior ao esforço requerido pela Híbrida. Estes tempos computacionais observados em geral confirmam o esperado: a execução do GRASP com a estratégia Fixa é a que demanda menos tempo, já que o valor de α próximo da escolha gulosa restringe as soluções construídas, que são em média de boa qualidade. Como conseqüência, o esforço da busca local é reduzido.

4.4 Problema de planarização de grafos

Para este problema, a estratégia Fixa utiliza $\alpha = 0,1$. Para a estratégia Híbrida, o valor de α é selecionado aleatoriamente a partir da seguinte distribuição de probabilidades, concentrada em torno da escolha gulosa: $p(\alpha_1 = 0,1) = 0,5$; $p(\alpha_2 = 0,2) = 0,25$; $p(\alpha_3 = 0,3) = 0,125$; $p(\alpha_4 = 0,4) = 0,03$; $p(\alpha_5 = 0,5) = 0,03$; $p(\alpha_6 = 0,6) = 0,03$; $p(\alpha_7 = 0,7) = 0,01$; $p(\alpha_8 = 0,8) = 0,01$; $p(\alpha_9 = 0,9) = 0,01$; $p(\alpha_{10} = 1) = 0,005$.

A exemplo do problema MAXSAT com pesos, o número de iterações efetuadas para a classe de instâncias tg não é suficiente para que se efetue uma comparação em termos da qualidade da solução aqui obtida com aquelas relatadas em [19]. Novamente, o desempenho do GRASP com variação do parâmetro α levou em média a melhores resultados do que a estratégia Fixa, como pode ser comprovado pelos resultados da Tabela 4.4. Em particular, para a classe de instâncias $cimi - g$, a solução ótima exata foi encontrada para todas as instâncias utilizando-se a variação do parâmetro, o que não foi possível para a instância cimi-g3 com α fixo.

Para a classe de instâncias do tipo g, o desempenho do GRASP com variação de parâmetro é claramente superior ao do GRASP com α fixo. Somente para a instância g12, com o número de iterações efetuadas, não foi possível encontrar a solução ótima global. Os tempos de processamento do GRASP com as quatro estratégias são praticamente equivalentes para as instâncias da classe g. Para a classe de instâncias $rg50$, a estratégia de variação de α ainda é a que permite a obtenção dos melhores resultados. Em particular, somente para a instância rg50.3 a estratégia Fixa superou o desempenho proporcionado pelas estratégias Reativa e Equiprovável. O esforço computacional é, para esta classe de instâncias, praticamente o mesmo para o GRASP com as quatro estratégias.

Com relação às instâncias $rg75$, os resultados não são conclusivos. Para a instância rg75.1, o melhor resultado foi obtido com a estratégia Fixa. Também para a classe de instâncias do tipo tg, os melhores resultados foram obtidos pela estratégia Fixa, com $\alpha = 0,1$. De acordo com [19], esta representa a melhor escolha para esta classe de instâncias. Este resultado não pode ser generalizado para as demais instâncias, conforme pode ser observado na Tabela 4.4. Aqui, a exemplo da classe de instâncias rg, os tempos computacionais são aproximadamente os mesmos para as quatro estratégias (a menos da Equiprovável, cujo esforço computacional é cerca de 26% superior àquele requerido pela Fixa).

5 Conclusões

Neste artigo, discutem-se estratégias de variação do parâmetro α em procedimentos GRASP e seu impacto no desempenho da meta-heurística, em termos da qualidade da solução obtida e do tempo de processamento requerido. As seguintes estratégias de variação de α foram consideradas: (i) α auto-ajustado ao longo da execução do algoritmo (GRASP Reativo); (ii) α selecionado a cada iteração aleatoriamente a partir de uma distribuição equiprovável de probabilidades; (iii) α selecionado aleatoriamente a cada iteração a partir de uma distribuição de probabilidade fixa, concentrada em torno do valor correspondente à escolha gulosa; e (iv) α fixo próximo da escolha gulosa.

As estratégias baseadas em variar o valor de α ao longo das iterações do GRASP introduzem um maior grau de diversificação nas soluções construídas na primeira fase do algoritmo, em relação à estratégia de se utilizar um mesmo valor em todas as iterações. Entretanto, levam em geral a tempos de processamento maiores, já que soluções de qualidade inferior também são construídas, o que torna mais lenta a busca local.

Os resultados obtidos mostram que a estratégia de se variar o valor de α ao longo das execuções do procedimento GRASP produz melhores soluções do que se considerar α fixo em todas as iterações. Além disso, dentre as estratégias de variação de α, a do GRASP Reativo constitui aquela que permite se obter as soluções de melhor qualidade. A estratégia de se utilizar um valor fixo próximo da escolha gulosa para α em todas as iterações produz resultados que são em média de boa qualidade, com um esforço computacional reduzido em relação às demais estratégias. Porém, comparativamente com as estratégias onde se varia o valor de α nas iterações do GRASP, esta jamais consegue superar as demais na qualidade das soluções obtidas, já que o grau de diversidade das soluções construídas é extremamente baixo.

Agradecimentos

Os autores agradecem a M.G.C. Resende pela cessão dos códigos Fortran utilizados nos testes efetuados com os problemas de recobrimento e MAXSAT com pesos, assim como a N. Paciornik pelas idéias e discussões sobre o procedimento GRASP Reativo.

Referências

[1] E. BALAS e P.R. LANDWEER, “Traffic assignment in communication satellites”, *Operations Research Letters* 2 (1983), 141-147.

[2] P. CAMERINI, F. MAFFIOLI e G. TARTARA, "Some scheduling algorithms for SS/TDMA systems", *Proceedings of the V International Conference on Digital Satellite Communications*, Genoa, 1981, 405-409.

[3] V. CHVÁTAL, "A greedy heuristic for the set covering problem", *Mathematics of Operations Research* 4 (1979), 233-235.

[4] R. CIMIKOWSKY, "An analysis of heuristics for the maximum planar subgraph problem", *Proceedings of the 6th ACM-SIAM Symposium on Discrete Algorithms*, 1995, 322-331.

[5] S. COOK, "The complexity theory-proving procedures", *Proceedings of the Third Annual ACM Symposium on Theory and Computing*, 1971, 151-158.

[6] M. DELLAMICO, F. MAFFIOLI e M. TRUBIAN, "New bounds for optimum traffic assignment in satellite communication", *Computers and Operations Research* 25 (1998), 729-743.

[7] T.A. FEO e M.G.C. RESENDE, "A probabilistic heuristic for a computationally difficult set covering problem", *Operations Research Letters* 8 (1989), 67-71.

[8] T.A. FEO e M.G.C. RESENDE, "Greedy randomized adaptive search procedures", *Journal of Global Optimization* 6 (1995), 109-133.

[9] D.R. FULKERSON, G.L. NEMHAUSER e L.E. TROTTER, JR., "Two computationally difficult set covering problems that arise in computing the 1-width of incidence matrices of Steiner triple systems", *Mathematical Programming Study* 2 (1974), 72-81.

[10] M.R. GAREY e D.S. JOHNSON, *Computers and intractability: A guide to theory of NP-completeness*, Freeman and Company, New York, 1979.

[11] O. GOLDSCHMIDT e A. TAKVORIAN, "An efficient graph planarization two-phase heuristic", *Networks* 24 (1994), 69-73.

[12] M. HALL, JR., *Combinatorial theory*, Blaisdell Company, Waltham, 1967.

[13] D. JOHNSON e M. TRICK, eds., *Cliques, Coloring, and Satisfatibility: Second DIMACS Implementation Challange*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, 1996.

[14] P. LIU e R. GELDMACHER, "On the deletion of nonplanar edges of a graph", *Proceedings of the 10th SE Conference on Combinatorics*, Graph Theory and Computing, Boca Raton, 1977, 727-738.

[15] M. MINOUX, "Optimal traffic assignment in a SS/TDMA frame: A new approach by set covering and column generation", *RAIRO Recherche Opérationnelle*

20 (1986), 1-13.

[16] M. PRAIS e C.C. RIBEIRO, "Reactive Grasp: An application to a matrix decomposition problem in TDMA traffic assignment", a ser publicado em *INFORMS Journal on Computing*, 1998.

[17] F. RENDL, "On the complexity of decomposing matrices arising in satellite communications", *Operations Research Letters* 4 (1985), 5-8.

[18] M.G.C. RESENDE, L.S. PITSOULIS e P.M. PARDALOS, "Approximate solution of weighted Max-Sat problems using GRASP", *DIMACS Series on Discrete Mathematics and Theoretical Computer Science* 35 (1997), 393-405.

[19] M.G.C. RESENDE e C.C. RIBEIRO, "A GRASP for Graph Planarization", *Networks* 29 (1997), 173-189.

[20] C.C. RIBEIRO, M. MINOUX e M.C. PENNA, "An optimal column-generation-with-ranking algorithm for very large scale set partitioning problems in traffic assignment", *European Journal of Operational Research* 41 (1989), 232-239.

[21] L. SCHRAGE, "A more portable Fortran random number generator", *ACM Transactions on Mathematical Software* 5 (1979) 132-138.

Problema	R		E		H		F		Melhor	Ótimo
	z^*	seg's	z^*	seg's	z^*	seg's	z^*	seg's	solução	?
P.06	565	6,7	565	7,0	565	6,9	565	6,1	Todas	Sim
P.07	3771	35,0	3771	25,0	3771	36,4	3771	43,0	Todas	Sim
P.08	4049	7,1	4049	7,6	4074	7,5	4149	6,3	R,E	Sim
P.09	3388	7,4	3388	9,2	3388	9,2	3443	6,3	R,E,H	Sim
P.10	3983	42,7	3983	31,1	3983	42,9	3983	39,8	Todas	Sim
P.11	3380	42,9	3380	26,9	3380	44,6	3380	47,0	Todas	Sim
P.12	657	32,4	657	22,8	657	32,8	657	33,5	Todas	Sim
P.13	3230	73,7	3230	41,8	3230	72,0	3245	90,6	R,E,H	Não
P.14	3157	86,4	3157	49,5	3157	89,8	3157	109,4	Todas	Sim
P.15	341	19,5	341	21,9	341	21,8	349	18,7	R,E,H	Sim
P.16	2343	139,9	2343	70,2	2343	134,3	2358	154,0	R,E,H	Sim
P.17	3281	19,9	3281	23,2	3281	23,4	3359	18,9	R,E,H	Sim
P.18	3228	25,9	3228	46,3	3228	64,0	3228	20,0	Todas	Sim
P.19	3710	128,5	3710	93,5	3710	136,9	3710	129,4	Todas	Sim
P.20	1860	111,1	1860	70,0	1860	121,8	1860	130,8	Todas	Não
P.21	3660	74,6	3660	71,0	3660	106,6	3660	88,5	R,E,H	Sim
P.22	1912	166,3	1912	124,7	1912	186,0	1925	207,8	R,E,H	Sim
P.23	3790	65,2	3770	68,2	3780	107,0	3810	94,0	E	Sim
P.24	661	265,0	661	189,0	661	267,0	661	277,2	Todas	Sim
P.25	504	153,8	504	120,9	504	181,9	504	173,6	Todas	Sim
P.26	520	188,4	520	139,3	520	237,2	520	237,2	Todas	Sim
P.27	216	45,8	216	47,4	216	47,5	216	44,8	Todas	Sim
P.28	1729	334,1	1729	221,1	1729	392,4	1729	369,4	Todas	Sim
P.29	3470	602,1	3470	444,6	3470	623,7	3470	657,1	Todas	Sim
P.30	4891	526,5	4891	324,7	4891	542,4	4902	543,8	R,E,H	Sim
P.31	620	688,6	620	557,6	620	682,1	620	696,5	Todas	Sim
P.32	2480	597,8	2480	331,6	2480	665,2	2480	716,6	Todas	Sim
P.33	3018	585,7	3018	394,6	3018	602,7	3018	608,8	Todas	Sim
P.34	1980	1172,9	1980	662,7	1980	1248,5	1980	1322,3	Todas	Sim
P.35	2140	1251,7	2140	736,5	2140	1287,7	2140	1341,4	Todas	Sim
P.36	7210	1554,0	7210	1313,8	7210	1583,3	7210	1551,0	Todas	Sim
P.37	6370	2112,7	6360	1263,1	6370	2186,9	6370	2210,5	E	Sim
P.38	2130	1451,5	2130	797,7	2130	1549,7	2130	1712,7	Todas	Sim
P.40	4984	2085,1	4984	1376,6	4984	2105,7	4984	2203,8	Todas	Não
P.41	2688	2879,7	2688	1584,5	2688	2916,6	2688	3207,0	Todas	Sim
P.42	2468	3261,9	2470	1587,0	2470	3687,2	2470	4048,8	R	Não
Média	-	579,0	-	358,2	-	612,6	-	642,8	-	-

Tabela 4.1: Resultados para o problema de decomposição de matrizes

Problema	R		E		H		F		Melhor	Ótimo
	z^*	seg's	z^*	seg's	z^*	seg's	z^*	seg's	solução	?
A_9	5	0,8	5	0,8	5	0,8	5	0,4	Todas	Sim
A_{15}	12	1,0	12	1,0	12	1,0	12	0,6	Todas	Sim
A_{27}	18	9,2	18	9,3	18	8,4	18	7,6	Todas	Sim
A_{45}	30	32,7	30	32,9	30	23,7	30	21,1	Todas	Sim
A_{81}	61	169,8	61	170,2	61	114,7	61	99,8	Todas	Sim
A_{135}	104	1035,0	105	1032,8	105	432,9	105	328,5	R	?
A_{243}	204	8179,1	204	8217,2	204	4096,0	205	3047,2	R,E,H	Não
Média	-	1346,8	-	1352,0	-	668,2	-	500,7	-	-

Tabela 4.2: Resultados para o problema de recobrimento

Problema	R		E		H		F		Melhor
	z^*	seg's	z^*	seg's	z^*	seg's	z^*	seg's	solução
jnh1	420769	2388,6	420769	2378,3	420780	1357,2	420650	1222,5	H
jnh4	420630	2494,6	420630	2497,9	420682	1738,4	420490	1617,0	H
jnh5	420556	2697,5	420556	2725,7	420626	1928,4	420626	1769,6	H,F
jnh6	420641	2633,4	420815	2642,0	420579	1885,9	420661	1798,2	E
jnh7	420925	178,6	420925	177,0	420925	795,8	420925	188,6	Todas
jnh8	420211	2799,3	420211	2768,9	419886	1879,1	419937	1757,2	R,E
jnh9	420115	2744,4	420115	269,7	419952	1650,1	420061	1645,6	R,E
jnh10	420485	2758,2	420531	2776,0	420404	1894,3	420501	1834,1	E
jnh11	420491	2502,7	420491	2521,4	420642	1718,2	420422	1607,2	H
jnh12	420920	2557,2	420920	2566,1	420871	1797,4	420871	1681,1	R,E
jnh13	420587	2697,4	420587	2715,5	420318	1897,5	420334	1781,1	R,E
jnh14	420696	2581,1	420559	2577,0	420574	1811,1	420593	1721,1	R
jnh15	420471	2739,7	420471	2715,1	420436	1918,5	420436	1799,3	R,E
jnh16	420883	2580,7	420883	2550,6	420817	1788,5	420817	1730,9	R,E
jnh17	420753	2458,9	420776	2493,2	420776	1452,7	420776	1302,5	E,H,F
jnh18	420497	2711,6	420497	2756,3	420537	1917,0	420604	1868,8	F
jnh19	420455	2493,2	420202	2574,6	420233	1825,2	420304	1779,8	R
Média	490593	2455,7	420584	2465,0	420531	1720,9	420530	1594,4	-
jnh201	394179	2084,3	394179	2144,6	394238	607,5	394238	946,3	H,F
jnh202	393983	2221,5	393983	2291,5	393899	1541,1	393901	1424,2	R,E
jnh203	393875	2483,4	393875	2503,9	393746	1859,6	393916	1698,3	F
jnh205	394063	2333,0	394063	2359,2	393959	1655,8	393959	1598,6	R,E
jnh207	394048	2194,4	394048	2217,4	394045	1518,3	394089	1445,4	F
jnh208	393863	2369,7	393863	2399,5	393997	1662,5	394098	1539,5	F
jnh209	394112	2251,8	394112	2267,0	394136	1610,8	394095	1566,2	H
jnh210	394238	765,8	394238	769,6	394238	1228,1	393907	1156,1	R,E,H
jnh211	393863	2502,7	393863	2518,9	393742	1849,9	393682	1765,5	R,E
jnh212	394115	2239,8	394115	2252,2	393999	1564,6	394227	1489,5	F
jnh214	393794	2353,5	393794	23430,9	393535	1732,1	393653	1641,8	R,E
jnh215	393942	2323,0	393942	2367,1	393942	1704,4	393878	1668,8	R,E,H
jnh216	394133	2500,4	394133	2524,1	394144	1794,2	393976	1699,4	H
jnh217	394218	2190,6	394218	2226,4	394219	1630,7	394224	1456,1	F
jnh218	394179	2256,6	394179	2327,4	394189	1633,0	394044	1575,0	H
jnh219	393686	2358,4	393720	2411,6	393635	1659,0	393646	1542,6	E
jnh220	394060	2145,0	394060	2203,4	394086	1970,0	394085	1386,8	H
Média	394021	2210,2	394023	2247,9	393985	1601,7	393978	1505,8	-
jnh301	444713	3002,0	444713	3051,7	444632	1970,0	444686	1866,9	R,E
jnh302	444305	2862,5	444305	2898,1	443896	2044,4	443946	1876,1	R,E
jnh303	444158	3050,9	444158	3084,8	444244	2220,3	444139	2155,0	H
jnh304	444334	2845,2	444334	2881,1	444427	1972,9	444071	1853,8	H
jnh305	443586	2998,9	443586	3013,8	443543	1968,1	443441	1810,3	R,E
jnh306	444829	2636,0	444829	2667,9	444515	1795,0	444515	1645,8	R,E
jnh307	444274	2860,9	444274	2916,2	444274	1899,6	444274	1779,1	Todas
jnh308	444345	2928,6	444345	2946,5	444275	2103,2	444155	1929,3	R,E
jnh309	444452	2857,5	444394	2893,5	444418	1986,2	444447	1820,3	R
jnh310	444135	3038,7	444135	3024,1	444073	2158,1	444146	2069,0	F
Média	444313	2908,1	444307	2957,8	444230	2011,7	444182	1880,6	-

Tabela 4.3: Resultados para o problema MAXSAT com pesos

Problema	R		E		H		F		Melhor	Ótimo
	z^*	seg's	z^*	seg's	z^*	seg's	z^*	seg's	solução	?
cimi-g1	19	22,2	19	22,1	19	21,8	19	21,2	Todas	Sim
cimi-g2	165	2088,3	165	2396,2	165	1962,8	165	1831,9	Todas	Sim
cimi-g3	73	313,8	73	290,9	73	249,3	64	245,6	R,E,H	Sim
cimi-g4	20	26,9	20	26,8	20	26,8	20	26,3	Todas	Sim
cimi-g5	82	448,7	82	455,9	82	450,8	82	456,0	Todas	Sim
cimi-g6	59	364,6	59	387,8	59	353,4	59	339,4	Todas	Sim
Média	-	544,1	-	596,6	-	510,8	-	486,7	-	-
g2	82	449,8	82	458,5	82	451,0	82	455,1	Todas	Sim
g3	24	0,0	24	0,0	24	0,0	23	37,5	R,E,H	Sim
g4	24	37,5	24	38,0	24	39,9	23	40,4	R,E,H	Sim
g5	24	42,0	24	41,8	24	44,4	23	44,5	R,E,H	Sim
g6	24	44,2	24	44,7	24	46,1	24	46,0	Todas	Sim
g7	24	67,0	24	67,0	24	65,0	24	64,1	Todas	Sim
g8	69	162,6	69	171,8	68	264,1	62	256,2	R,E	Sim
g9	69	300,1	69	317,0	69	271,1	62	264,2	R,E,H	Sim
g10	69	292,0	69	309,0	69	247,2	61	225,4	R,E,H	Sim
g11	69	304,0	69	322,4	68	256,8	61	233,2	R,E	Sim
g12	67	524,3	68	556,6	67	474,3	65	444,4	E	Não
Média	-	202,1	-	211,5	-	196,4	-	191,9	-	-
rg50.1	90	1329,3	90	1419,5	90	1268,8	90	1297,6	Todas	?
rg50.2	95	1705,6	95	1791,2	95	1633,1	95	1585,8	Todas	?
rg50.3	102	2249,6	102	2380,7	103	2169,8	103	2131,2	H,F	?
rg50.4	103	2791,5	103	2899,7	103	2748,4	101	2754,6	R,E,H	Não
rg50.5	106	3315,3	105	3422,5	106	3309,1	105	3378,5	R,H	?
Média	-	2278,3	-	2382,7	-	2225,8	-	2229,5	-	-
rg75.1	128	3897,1	128	4199,2	129	3687,3	130	3564,4	F	Não
rg75.2	131	4000,3	131	4483,1	131	3813,2	131	3601,3	Todas	?
rg75.3	136	4896,2	134	5366,7	134	4693,4	134	4575,0	R	?
rg75.4	143	7705,6	142	8182,0	143	7395,0	143	7248,3	R,H,F	?
rg75.5	145	8746,0	142	9280,4	144	8289,6	143	8040,1	R	?
Média	-	5849,0	-	6302,3	-	5575,7	-	5405,8	-	-
tg100.1	262	12527,4	256	15000,3	262	12579,3	258	11635,7	R,H	Não
tg100.2	257	14030,6	256	17091,5	257	14027,9	265	12781,0	F	Não
tg100.3	266	14705,6	263	17507,6	266	14809,7	264	13875,1	R,H	Não
tg100.4	263	16468,6	257	19834,1	261	16670,7	262	15399,6	R	Não
tg100.5	257	18042,4	253	21798,6	256	18232,4	259	16953,8	F	Não
tg100.6	251	19390,8	250	22673,6	251	19332,0	253	18240,2	F	Não
tg100.7	252	21198,7	247	24994,4	252	21192,9	246	19905,5	R,H	Não
tg100.8	250	22516,3	250	26149,7	250	22745,7	252	21420,5	F	Não
tg100.9	245	24097,8	242	28173,8	246	24039,2	249	22755,9	F	Não
tg100.10	241	26333,3	237	30265,9	242	26219,0	247	24789,1	F	Não
Média	-	18931,2	-	22349,0	-	18984,9	-	17775,6	-	-

Tabela 4.4: Resultados para o problema de planarização de grafos