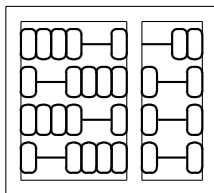


UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO



MO420/MC908: Tópicos em Otimização Combinatória
Prof. Cid Carvalho de Souza - IC/UNICAMP
2o Trabalho Prático – 2o. semestre de 2012

RESOLUÇÃO DO PROBLEMA *Maximum 2-Independent Sets* UTILIZANDO
BRANCH-AND-CUT

Aluno: Diego Rodrigo Hachmann - RA 134047

RA: Idogor Rosembert - RA 134058

CAMPINAS

2012

1 O Problema Maximum 2-independent Sets

Este trabalho é fortemente baseado no artigo de Nemhauser e Sigismondi ¹. Heurísticas, rotinas de separação e geração dos cliques iniciais, utilizadas neste trabalho, podem ser consultadas na referência acima citada.

1.1 Descrição do Problema

Um conjunto independente de um grafo \mathcal{G} é um conjunto de vértices S tal que não existem dois vértices adjacentes contidos em S . O problema Maximum 2-independent Sets (2IS) consiste em maximizar a soma das cardinalidades de dois conjuntos de vértices independentes com intersecção vazia do grafo \mathcal{G} .

1.2 Formulações

Uma formulação natural(FN) em programação linear é dada abaixo:

$$\max z = \sum_{u \in V} x_{Au} + \sum_{u \in V} x_{Bu} \quad (1)$$

$$\text{s.a.} : x_{Au} + x_{Av} \leq 1, \quad \forall (u, v) \in E \quad (2)$$

$$x_{Bu} + x_{Bv} \leq 1, \quad \forall (u, v) \in E \quad (3)$$

$$x_{Au} + x_{Bu} \leq 1, \quad \forall u \in V \quad (4)$$

$$x_{Au} \in \{0, 1\} \text{ e } x_{Bu} \in \{0, 1\} \quad \forall u \in V \quad (5)$$

Sabe-se que a formulação FN é ruim no sentido de que sua relaxação linear fornece limitantes duais fracos, fazendo com que algoritmos de *branch-and-bound* convencionais encontrem dificuldade ao resolver instâncias não patológicas com 40 nós ou até mesmo menores.

O problema 2IS pode ser formulado em função das cliques maximais de \mathcal{G} (a partir daqui chamaremos clique maximal apenas como clique). É fácil perceber que um conjunto independente contém não mais que um nó em cada clique de \mathcal{G} . Assim, temos a seguinte formulação por cliques (CF):

¹<http://www.jstor.org/stable/2583564>

$$\max z = \sum_{u \in V} x_{Au} + \sum_{u \in V} x_{Bu} \quad (6)$$

$$\text{s.a : } \sum_{v \in C} x_{Av} \leq 1, \text{ para toda clique maximal } C \quad (7)$$

$$\sum_{v \in C} x_{Bv} \leq 1, \text{ para toda clique maximal } C \quad (8)$$

$$x_{Au} + x_{Bv} \leq 1, \quad \forall u, v \in V \quad (9)$$

$$x_{Au} \in \{0, 1\} \text{ e } x_{Bv} \in \{0, 1\}; \quad \forall u, v \in V \quad (10)$$

na qual cada aresta de \mathcal{G} está em pelo menos uma clique. Não são criadas restrições para todas as cliques C , pois para qualquer entrada razoável o número de cliques provavelmente é exponencial. Ao invés disso, são escolhidos alguns cliques maximais que cubram todas as arestas para o modelo inicial, sem que mais cliques podem ser adicionados na rotina de separação por cortes.

A relaxação linear do modelo CF fornece limitantes ao menos tão bons quanto ou em geral muito melhores que os obtidos com a relaxação linear do modelo FN, visto que todas as restrições de cliques definem facetas da envoltória convexa das soluções do I2S.

1.3 Descrição das desigualdades

1.3.1 Modelo FN

As restrições (2) e (3) garantem que dois nós adjacentes u e v não sejam adicionados a um mesmo conjunto independente. As restrições (4) garantem que um vértice pertença a no máximo um conjunto independente.

É fácil notar que as desigualdades (2) e (3) são válidas, pois elas são derivadas diretamente da definição de conjunto independente. As restrições (4) são válidas pois pela definição do Problema 2IS, um nó pode fazer parte de no máximo um conjunto independente, isto é, a interseção entre os dois conjuntos deve ser vazia.

1.3.2 Modelo CF

As restrições (7) e (8) garantem que uma clique não possua mais que um nó no mesmo conjunto independente. As restrições (9) garantem que um vértice do grafo esteja apenas em um dos conjuntos independentes.

As desigualdades (7) e (8) são válidas. Se uma clique possuir mais que um nó de um mesmo conjunto independente I , então tais nós são adjacentes e portanto não podem simultaneamente fazer parte de I . As restrições (9) é válida pois pela definição do Problema 2IS, um nó pode fazer parte de no máximo um conjunto independente.

1.4 Dimensão do Poliedro

Proposição 1. *A dimensão do poliedro descrito pela formulação FN é $2 * |V|$*

Demonstração. Basta provar que o poliedro possui $2 * |V| + 1$ vetores afim independentes. Sejam A e B os dois conjuntos independentes. Seja S o conjunto de soluções viáveis para a formulação FN:

$$S = v_A \forall v_A \in V \cup v_B \forall v_B \in V$$

Ou seja, S é o conjunto de soluções com apenas um vértice no conjunto A e nenhum vértice no conjunto B unido com as soluções com apenas um vértice no conjunto B e nenhum no conjunto A. Seja A a matriz formada por todos os vetores característicos $\chi^s \forall s \in S$. É fácil perceber que A é uma matriz identidade, logo o número de linhas linearmente independentes é $2 * |V|$. Temos ainda que χ^0 (vetor nulo) também é uma solução para o problema 2IS e portanto temos $2 * |V| + 1$ vetores afim independentes. \square

Podemos usar o mesmo argumento dado na prova acima para provar que a formulação CF também possui dimensão $2 * |V|$.

2 Geração dos cliques da formulação - CLQ1

A rotina CLQ1, apresentada no Algoritmo 1 gera as cliques iniciais da formulação CF tal que toda a aresta do grafo de entrada seja coberta por pelo menos uma clique.

Data: Grafo $G = (V, E)$

Result: Conjunto de cliques maximais que cobrem todas as arestas de G

C = conjunto de cliques, inicialmente vazio;

T = conjunto de arestas já selecionadas, inicialmente vazio;

```
while  $T \neq E$  do
     $e = \text{alguma aresta} \in E \setminus T$ 
     $T = T \cup \{e\}$ 
     $Q = \{e.u, e.v\}$ 
    while puder aumentar  $Q$  do
         $v = \text{vértice adjacente a todos os vértices em } Q$ 
         $W = \text{conjunto de arestas adjacentes a } v \text{ e todos os vértices de } Q$ 
         $Q = Q \cup \{v\}$ 
         $T = T \cup W$ 
    end
     $C = C \cup \{Q\}$ 
end

retorne  $C$ 
```

Algorithm 1: Algoritmo CLQ1

2.1 Rotinas de Separação

Dada uma solução ótima para a relaxação de um programa linear inteiro, uma rotina de separação tenta achar uma desigualdade válida para a formulação inteira que é violada pela solução ótima da relaxação.

Seja L o conjunto de todas as cliques encontrados pelo procedimento CLQ1. A rotina de separação implementada tenta achar uma clique $C \notin L$ com $\sum_{v \in C} > 1$. Uma vez que a solução relaxada é inviável se tal C não existe, $C \notin L$. Ao invés de adicionarmos no máximo um único corte a cada momento que o cplex chama o callback da rotina de corte, resolvemos adicionar no máximo cinco cortes. Além disso, só é adicionado os cortes se a altura do nó da relaxação linear é não for maior que 10.

Os algoritmos 2 e 3 são utilizadas para a separação. Testes realizados mostraram que o tempo das rotinas de corte são muito baratas se comparado ao tempo total da solução do programa linear. Decidimos então executar a rotina mais vezes para encontrar no máximo 50 cortes e destes selecionar apenas os cinco que mais violam a solução ótima fracionária. Outro critério do algoritmo é que a interseção de todos os cortes são vazios, ou seja, uma variável só pode aparecer em apenas um corte.

Data: Grafo G , Solução Relaxada x^*

Result: Conjunto de Cortes que violam x^*

if *Profundida do nó atual* > 10 **then**

 | retorne \emptyset ;

end

C = conjunto de cortes, inicialmente vazio

V = conjunto de vértices já em algum corte, inicialmente vazio

for $i \in 1 \dots 50$ **do**

for *cada conjunto independente* S **do**

$x^{**} = x^* \mid A \#$ parte da solução referente ao conjunto S

 Corte $a = \text{Corte}(G, x^{**}, CLQ2A, R)$

 Corte $b = \text{Corte}(G, x^{**}, CLQ2B, R)$

if $a=b=$ "*Todos vértices utilizados*" **then**

 | goto fim

end

$C = a \cup b$ #somente se forem válidos;

end

end

fim: retorne os 5 cortes em C que mais violam x^*

Algorithm 2: Algoritmo NCortes

Data: Grafo G , Solução Relaxada x^* , Critério de Seleção CLQ2, Conjunto de vértices R já escolhidos
em outro corte para a mesma solução x^*

Result: Clique máximo S , que viola x^* , se existir

S = clique inicialmente vazio

while *conseguir aumentar S* **do**

$x = \text{maximiza } x_i \in x^* \text{ conforme critério CLQ2 AND } \notin R \text{ AND adjacente a todos os vértices em } S$
 $R = R \cup \{x\}$
 $S = S \cup \{x\}$

end

if $S = \emptyset$ **then**

 retorne "Todos vértices utilizados";

end

sum = soma dos valores dos vértices em S ;

if $sum > 1 + EPS$ **then**

 retorne corte $\sum_{v \in S} x_v \leq 1$;

else

 retorne "corte não encontrado";

end

Algorithm 3: Algoritmo Corte

Os dois critérios de escolha utilizados nos algoritmos 2 e 3 são:

- CLQ2A : Escolha uma variável que maximize $\{x_v < 1\}$
- CLQ2B : Escolha uma variável que minimize $\{|x_v - 1/2| : 0 < x_v < 1\}$

3 Resultados

Foram utilizadas 60 instâncias de testes. O nome de cada instância inicia com a letra g e o número de nós do grafo descrito por ela. A instância $g60_{10.1}$, por exemplo, possui 60 vértices. O tempo máximo de execução de cada instância foi de 20 minutos (1200 segundos). Os experimentos foram executados em um servidor com processador Intel Xeon 2.67GHz com 42GB de memória RAM. O cplex foi configurado para utilizar o tempo de CPU ao invés do tempo de máquina, impedindo que outros processos concorrentes pudessem interferir na qualidade dos dados obtidos.

Tabela 1: Resultados para densidade 0.10

	FN 1200 1				CQLBB 1200 1				CLQBC 1200 1			
I1	C1	C2	D1	D2	C1	C2	D1	D2	C1	C2	D1	D2
g60_10_1	43	0	3	9376	43	0	0	631	43	0	0	631
g60_10_2	43	0	5	10213	43	0	0	776	43	0	0	496
g60_10_3	42	0	8	19554	42	0	0	296	42	0	0	296
g80_10_1	52	0	176	277501	52	0	4	3718	52	0	4	2502
g80_10_2	50	0	545	908836	50	0	15	12721	50	0	21	10719
g80_10_3	52	0	249	370691	52	0	10	9771	52	0	15	9225
g100_10_1	57	12	1205	1513210	58	0	387	193643	58	0	553	230178
g100_10_2	55	16	1204	1512504	57	1	1201	646557	57	1	1202	527593
g100_10_3	56	14	1205	1512072	57	0	954	424801	57	0	630	254859
g120_10_1	60	29	1205	1002844	62	8	1203	298080	61	9	1203	233820
g120_10_2	61	28	1204	1120187	64	5	1202	243761	65	4	1202	193539
g120_10_3	60	29	1204	1086153	64	4	1202	262071	63	5	1203	232916

Tabela 2: Resultados para densidade 0.30

	FN 1200 1				CQLBB 1200 1				CQLBC 1200 1			
I1	C1	C2	D1	D2	C1	C2	D1	D2	C1	C2	D1	D2
g60_30_1	24	0	180	237872	24	0	2	1304	24	0	4	849
g60_30_2	23	0	281	412555	23	0	4	3077	23	0	6	1483
g60_30_3	23	0	258	390342	23	0	3	2565	23	0	4	893
g80_30_1	25	13	1204	1194832	26	0	150	52348	26	0	109	12082
g80_30_2	25	12	1204	1166355	27	0	61	20335	27	0	56	5599
g80_30_3	26	10	1204	1179326	27	0	158	48308	27	0	155	16628
g100_30_1	27	27	1203	822147	30	0	738	125786	30	0	458	32071
g100_30_2	26	27	1202	828584	29	1	1202	208478	28	3	1201	70956
g100_30_3	25	29	1202	803526	28	3	1202	232443	28	3	1202	84629
g120_30_1	25	48	1202	516271	30	8	1204	149932	30	7	1203	46936
g120_30_2	26	46	1203	511231	28	10	1202	151564	29	8	1203	58022
g120_30_3	27	47	1203	481929	27	11	1204	156435	29	9	1204	56601

3.1 Formulações

Nesta seção comparamos o desempenho das formulações FN (modelo FN) e CLQ1 (modelos CLBQQ e CLQBC). Todas os testes foram executados utilizando as heurísticas implementadas. As instâncias foram agrupadas pela densidade dos grafos de entrada. Os resultados podem ser vistos nas tabelas 1, 2, 3, 4 e 5.

C1: Maior solução primal encontrada

C2: Menor GAP (solução dual - valor do inbumbent)

A seguinte notação foi utilizada:

D1: Tempo em segundos

D2: Número de iterações

Como era de se esperar, os resultados encontrados para a formulação FN foram insatisfatórios quando comparado a formulação CLQ1, obtendo apenas 18 resultados ótimos dentre os 60 possíveis, contra 46 do

Tabela 3: Resultados para densidade 0.50

	FN 1200 1				CQLBB 1200 1				CQLBC 1200 1			
I1	C1	C2	D1	D2	C1	C2	D1	D2	C1	C2	D1	D2
g60_50_1	14	0	258	305629	14	0	4	2339	14	0	11	998
g60_50_2	15	0	144	151257	15	0	6	2715	15	0	7	717
g60_50_3	14	0	233	276000	14	0	8	3824	14	0	10	1047
g100_50_1	17	24	1202	558093	18	0	484	65998	18	0	541	15730
g100_50_2	17	26	1202	415451	18	0	489	72719	18	0	533	15769
g100_50_3	16	28	1203	399199	18	0	499	66049	18	0	636	19492
g120_50_1	17	44	1201	317890	17	7	1202	104925	19	5	1204	24576
g120_50_2	17	44	1201	236306	19	5	1202	102116	19	5	1203	19717
g120_50_3	17	47	1202	173774	18	6	1202	99911	17	7	1200	19162
g80_50_1	17	6	1203	818171	17	0	58	16885	17	0	32	1823
g80_50_2	16	8	1202	833316	16	0	52	13134	16	0	95	4543
g80_50_3	16	8	1203	805574	16	0	73	16286	16	0	106	6230

Tabela 4: Resultados para densidade 0.70

	FN 1200 1				CQLBB 1200 1				CQLBC 1200 1			
I1	C1	C2	D1	D2	C1	C2	D1	D2	C1	C2	D1	D2
g60_70_1	11	0	63	52657	11	0	2	371	11	0	2	30
g60_70_2	11	0	46	41480	11	0	1	308	11	0	2	17
g60_70_3	10	0	74	63980	10	0	2	534	10	0	4	130
g80_70_1	11	0	613	330992	11	0	18	3953	11	0	34	923
g80_70_2	12	0	480	220608	12	0	7	1297	12	0	13	279
g80_70_3	12	0	452	214169	12	0	15	2738	12	0	16	433
g100_70_1	12	21	1201	223765	12	0	158	24341	12	0	143	3742
g100_70_2	11	21	1202	220028	12	0	179	22989	12	0	112	2745
g100_70_3	11	22	1202	242839	12	0	118	14694	12	0	114	3593
g120_70_1	12	42	1201	144017	13	0	563	34791	13	0	436	5914
g120_70_2	12	40	1201	139830	12	2	1201	68461	12	1	1201	15126
g120_70_3	11	41	1202	148093	12	0	1086	70790	12	2	1200	14824

Tabela 5: Resultados para densidade 0.90

	FN 1200 1				CQLBB 1200 1				CQLBC 1200 1			
I1	C1	C2	D1	D2	C1	C2	D1	D2	C1	C2	D1	D2
g60_90_1	6	0	27	13904	6	0	0	434	6	0	4	0
g60_90_2	6	0	19	14579	6	0	0	148	6	0	0	0
g60_90_3	7	0	17	11932	7	0	0	314	7	0	0	0
g80_90_1	7	0	115	34851	7	0	3	998	7	0	13	0
g80_90_2	8	0	101	30049	8	0	3	690	8	0	6	0
g80_90_3	8	0	137	31152	8	0	3	822	8	0	1	0
g120_90_1	8	19	1201	98648	8	0	130	7915	8	0	185	117
g100_90_2	8	0	433	70136	8	0	23	2851	8	0	39	0
g100_90_3	8	0	545	78380	8	0	12	2125	8	0	54	65
g100_90_1	8	0	395	66491	8	0	20	3167	8	0	26	0
g120_90_2	8	17	1203	101512	8	0	89	6510	8	0	153	422
g120_90_3	8	8	1201	109607	8	0	62	4538	8	0	127	91

[!H!]

da formulação CLQ1. Para os as instâncias que não obtiveram resultados ótimos, o erro encontrado na formulação FN foi bem superior, chegando a casa dos 100%.

Podemos observar que quanto maior a densidade do grafo, melhor é a qualidade da solução encontrada e menor é o número de nós explorados. Ainda, é possível notar que a formulação CLQBB tende ser mais eficiente que a formulação CLQBC a medida que a densidade do grafo aumenta.

3.2 Heurísticas

Nesta seção são analisados os resultados obtidos com as heurísticas implementadas neste trabalho. Para tanto, comparamos o desempenho do algoritmo branch-and-bound -CLQBB- com e sem o uso das heurísticas. Os resultados são apresentados na Tabela 6. A coluna H1 mostra a quantidade de heurísticas aplicadas que melhoraram o *incumbent*. As outras colunas foram definidas anteriormente. Os valores em verde significam que o valor obtido com o uso da heurística foi melhorado (menor tempo, menor GAP, menor número de nós explorados) quando comparado ao não uso das heurísticas. Os valores em verdes tem significado contrário.

Tabela 6: Resultado do uso das heurísticas

	Sem heurísticas			Com heurística			
I1	D1	G1	D2	H1	D1	G1	D2
g100_10_1	414	0	193950	7	387	0	193643
g100_10_2	1202	2	530611	5	1201	1	646557
g100_10_3	591	0	252958	5	954	0	424801
g100_30_1	646	0	110660	1	738	0	125786
g100_30_2	1200	3	178597	3	1202	1	208478
g100_30_3	1202	3	207017	5	1202	3	232443
g100_50_1	573	0	83146	1	484	0	65998
g100_50_2	451	0	64173	3	489	0	72719
g100_50_3	521	0	71397	2	499	0	66049
g100_70_1	146	0	21182	3	158	0	24341
g100_70_2	158	0	18125	3	179	0	22989
g100_70_3	127	0	15941	1	118	0	14694
g100_90_1	17	0	2538	2	20	0	3167
g100_90_2	26	0	2780	3	23	0	2851
g100_90_3	11	0	1877	1	12	0	2125
g120_10_1	1201	8	270087	2	1203	8	298080
g120_10_2	1201	5	210085	6	1202	5	243761

g120_10_3	1202	4	257237	3	1202	4	262071
g120_30_1	1202	9	136229	3	1204	8	149932
g120_30_2	1203	10	151501	2	1202	10	151564
g120_30_3	1203	10	147254	2	1204	11	156435
g120_50_1	1203	5	108559	2	1202	7	104925
g120_50_2	1202	4	94896	3	1202	5	102116
g120_50_3	1203	4	101366	3	1202	6	99911
g120_70_1	559	0	40099	3	563	0	34791
g120_70_2	1201	1	80075	4	1201	2	68461
g120_70_3	865	0	60141	3	1086	0	70790
g120_90_1	107	0	8025	2	130	0	7915
g120_90_2	62	0	5806	1	89	0	6510
g120_90_3	49	0	4451	3	62	0	4538
g60_10_1	0	0	583	3	0	0	631
g60_10_2	0	0	490	2	0	0	776
g60_10_3	0	0	322	2	0	0	296
g60_30_1	3	0	1755	2	2	0	1304
g60_30_2	4	0	3465	2	4	0	3077
g60_30_3	5	0	2247	2	3	0	2565
g60_50_1	5	0	2721	2	4	0	2339
g60_50_2	6	0	2735	2	6	0	2715
g60_50_3	6	0	3471	2	8	0	3824
g60_70_1	1	0	227	1	2	0	371
g60_70_2	1	0	446	3	1	0	308
g60_70_3	1	0	575	2	2	0	534
g60_90_1	0	0	346	2	0	0	434
g60_90_2	0	0	145	2	0	0	148
g60_90_3	0	0	298	1	0	0	314
g80_10_1	6	0	4811	3	4	0	3718
g80_10_2	14	0	10932	5	15	0	12721
g80_10_3	11	0	8870	5	10	0	9771
g80_30_1	148	0	53071	4	150	0	52348
g80_30_2	51	0	14752	4	61	0	20335
g80_30_3	190	0	60118	1	158	0	48308

g80_50_1	41	0	11091	2	58	0	16885
g80_50_2	60	0	14174	3	52	0	13134
g80_50_3	68	0	16808	2	73	0	16286
g80_70_1	14	0	3241	1	18	0	3953
g80_70_2	10	0	2304	4	7	0	1297
g80_70_3	13	0	3369	3	15	0	2738
g80_90_1	3	0	1116	2	3	0	998
g80_90_2	2	0	525	2	3	0	690
g80_90_3	2	0	791	2	3	0	822

Da coluna H1, concluímos que, independente do numero de nós explorados (coluna D2), a quantidade de *incumbents* melhorados não sofreu aumento significativo. Quanto a melhora da solução pelo uso das heurísticas, percebe-se que não houve uma comportamento padrão de melhora.

Observando o log do cplex, percebemos que na maioria das instâncias o BB com o uso das heurísticas proporcionou melhora no *incumbent* logo no inicio da execução. Por outro lado, o BB sem heurísticas encontra mais soluções inteiras alguns segundos após o inicio da execução.

3.3 Cortes

Nesta seção são analisados os resultados obtidos com a rotina de separação por cortes implementadas neste trabalho. Para tanto, comparamos o desempenho do algoritmo branch-and-bound -CLQBC- com o algoritmo branch-and-cut -CLQBC-, ambos com o uso das heurísticas. Os resultados são apresentados na Tabela 7. A coluna N1 mostra a quantidade de cortes aplicados. As outras colunas foram definidas anteriormente. Os valores em verde significam que o valor obtido com o uso dos cortes foi melhorado (menor tempo, menor GAP, menor número de nos explorados) e em vermelho piorou.

Tabela 7: Resultado do uso dos cortes

I1	Sem cortes			Com cortes			
	D1	G1	D2	N1	D1	G1	D2
g100_10_1	387	0	193643	24	556	0	230178
g100_10_2	1201	1	646557	8	1202	1	527656
g100_10_3	954	0	424801	22	624	0	254859
g100_30_1	738	0	125786	963	458	0	32071
g100_30_2	1202	1	208478	1057	1201	3	70956

g100_30_3	1202	3	232443	1095	1202	3	84629
g100_50_1	484	0	65998	2156	541	0	15730
g100_50_2	489	0	72719	2474	533	0	15769
g100_50_3	499	0	66049	2037	636	0	19492
g100_70_1	158	0	24341	1943	143	0	3742
g100_70_2	179	0	22989	1601	112	0	2745
g100_70_3	118	0	14694	1592	114	0	3593
g100_90_1	20	0	3167	697	26	0	0
g100_90_2	23	0	2851	729	39	0	0
g100_90_3	12	0	2125	1191	54	0	65
g120_10_1	1203	8	298080	65	1203	9	235207
g120_10_2	1202	5	243761	24	1203	4	194976
g120_10_3	1202	4	262071	42	1203	5	241406
g120_30_1	1204	8	149932	1582	1203	7	46936
g120_30_2	1202	10	151564	1423	1203	8	58022
g120_30_3	1204	11	156435	1450	1204	9	56601
g120_50_1	1202	7	104925	2702	1204	5	24576
g120_50_2	1202	5	102116	2981	1203	5	19717
g120_50_3	1202	6	99911	3139	1200	7	19162
g120_70_1	563	0	34791	2235	436	0	5914
g120_70_2	1201	2	68461	2786	1201	1	15126
g120_70_3	1086	0	70790	2460	1200	2	14824
g120_90_1	130	0	7915	1362	185	0	117
g120_90_2	89	0	6510	1703	153	0	422
g120_90_3	62	0	4538	1344	127	0	91
g60_10_1	0	0	631	0	0	0	631
g60_10_2	0	0	776	2	0	0	496
g60_10_3	0	0	296	0	0	0	296
g60_30_1	2	0	1304	197	4	0	849
g60_30_2	4	0	3077	228	6	0	1483
g60_30_3	3	0	2565	176	4	0	893
g60_50_1	4	0	2339	756	11	0	998
g60_50_2	6	0	2715	537	7	0	717
g60_50_3	8	0	3824	718	10	0	1047

g60_70_1	2	0	371	292	2	0	30
g60_70_2	1	0	308	296	2	0	17
g60_70_3	2	0	534	580	4	0	130
g60_90_1	0	0	434	303	4	0	0
g60_90_2	0	0	148	217	0	0	0
g60_90_3	0	0	314	90	0	0	0
g80_10_1	4	0	3718	6	5	0	2502
g80_10_2	15	0	12721	4	19	0	10719
g80_10_3	10	0	9771	4	15	0	9225
g80_30_1	150	0	52348	597	109	0	12082
g80_30_2	61	0	20335	578	56	0	5599
g80_30_3	158	0	48308	553	155	0	16628
g80_50_1	58	0	16885	1113	32	0	1823
g80_50_2	52	0	13134	1216	95	0	4543
g80_50_3	73	0	16286	1185	106	0	6230
g80_70_1	18	0	3953	1127	34	0	923
g80_70_2	7	0	1297	1152	13	0	279
g80_70_3	15	0	2738	1114	16	0	433
g80_90_1	3	0	998	451	13	0	0
g80_90_2	3	0	690	446	6	0	0
g80_90_3	3	0	822	196	1	0	0

Claramente a coluna D2 mostra que o uso dos cortes de cliques maximais proporcionou uma diminuição no número de nós explorados, apesar de a diminuição no tempo de execução nem sempre ser observada. Novamente observamos no log do cplex que as melhores com o uso de cortes eram obtidas logo no início da execução, ficando mais lenta depois de um certo tempo.

4 Conclusão

Neste trabalho foram implementados rotinas de separação de cortes e heurísticas que convertem uma solução fracionário em uma solução inteira do problema 2IS. Dois modelos foram implementados, FN e CLQ, sendo que o segundo resultados muito superiores ao primeiro. Foram realizados experimentos com os algoritmos de branch-and-bound e branch-and-cut implementados na biblioteca cplex. Os experimentos mostram que instâncias com grafos mais densos são mais "fáceis" de serem solucionadas. Os experimentos mostraram ainda

que as heurísticas não proveram grandes melhorias nos resultados. Os cortes de cliques maximais diminuíram o número de nós explorados, porém como cortes eram adicionados, não houve melhora perceptiva nos tempos de execução.