Using PMaxSat Techniques to Solve the Maximum Clique Problem

Alexandre Prusch Züge Renato Carmo
Ricardo Tavares de Oliveira Fabiano Silva

Universidade Federal do Paraná
Programa de Pós-Graduação em Informática
Grupo de Pesquisa em Algoritmos

November 10, 2016

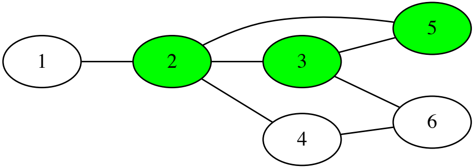
7 LAWCG
LA PLATA, ARGENTINA 2016

UFPR

Introduction

Maximum Clique Problem (MC)
Input : Simple graph G.
Output: Maximum clique in G.

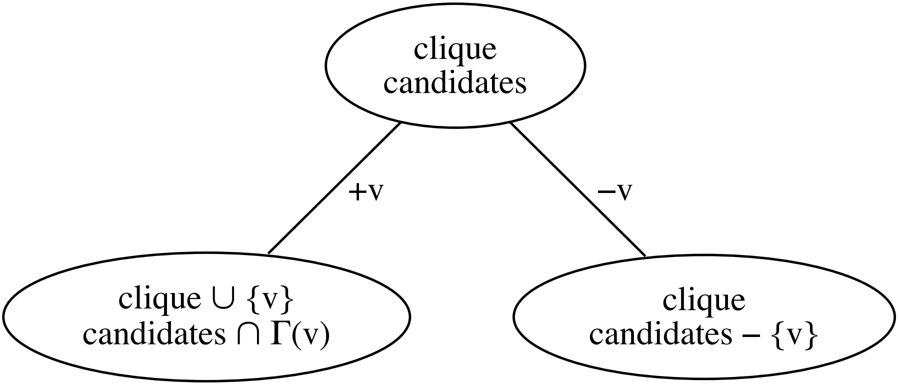
Maximum clique: $\{2, 3, 5\}$.



Clique: set of vertices inducing a complete subgraph.

Branch and Bound Scheme for MC

Generating cliques via backtracking.



PREVIOUS WORK

- In Li and Quan (2010a), a reduction from MC to Partial Maximum Satisfiability problem (PMAXSAT) was proposed.
- The authors stated that using a solver was not competitive against dedicated algorithms for MC, but were able to "borrow" some techniques from a PMAXSAT solver and improve a dedicated algorithm.
- This work is explained in ZÜGE, A.P.; CARMO, R. Maximum Clique via MaxSat and Back Again. *Matemática Contemporânea.* Vol. 44. 2015.

Previous Work

- Li and Quan (2010b)
- Li, Fang, and Xu (2013)
- Li, Jiang, and Xu (2015)
- Segundo, Nikolaev, and Batsyn (2015)

Relation to Partial MaxSat

Definitions

- literal: a boolean variable or its negation
- clause: disjunction of literals
- assignment: set of literals where no variable can be repeated

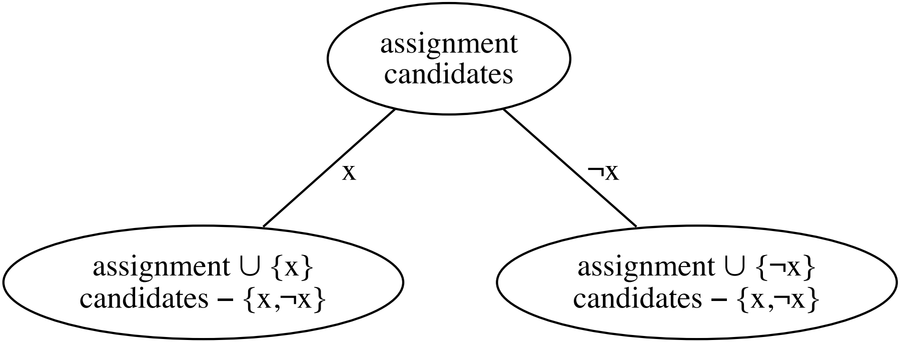
Partial Maximum Satisfiability (PMaxSat)

Input : Set of variables, set of hard clauses and set of soft clauses.

Output: Assignment of the variables satisfying all hard clauses and the maximum number of soft clauses.

Branch and Bound Scheme for PMaxSat

Generating assignments via backtracking.

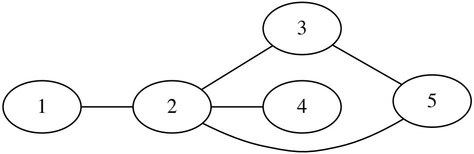


MC → Partial MaxSat: Usual Reduction

- A variable x for each vertex $v \in V(G)$.
- Hard clauses: $\{(\neg x_i \vee \neg x_j) : \{v_i, v_j\} \notin E(G)\}$
- Soft clauses: $\{(x_i) : v_i \in V(G)\}$

MC → Partial MaxSat: Usual Reduction

- A variable x for each vertex $v \in V(G)$.
- Hard clauses: $\{(\neg x_i \vee \neg x_j) : \{v_i, v_j\} \notin E(G)\}$
- Soft clauses: $\{(x_i) : v_i \in V(G)\}$



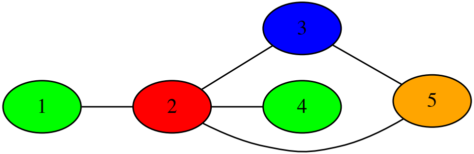
Hard clauses: $\{(\neg 1 \vee \neg 3), (\neg 1 \vee \neg 4), (\neg 1 \vee \neg 5), (\neg 3 \vee \neg 4), (\neg 4 \vee \neg 5)\}$
Soft clauses: $\{(1), (2), (3), (4), (5)\}$

MC → Partial MaxSat: Li and Quan (2010a) Reduction

- A variable x for each vertex $v \in V(G)$.
- Hard clauses: $\{(\neg x_i \vee \neg x_j) \colon \{v_i, v_j\} \notin E(G)\}$
- Soft clauses: $\{\bigvee_{x_i \in C} x_i \colon C \in \gamma(G)\}$
  - Computes a coloring $\gamma(G)$.
  - One soft clause for each color.

MC $\rightarrow$ Partial MaxSat: Li and Quan (2010a) Reduction

- A variable x for each vertex $v \in V(G)$.
- Hard clauses: $\{(\neg x_i \vee \neg x_j) : \{v_i, v_j\} \notin E(G)\}$
- Soft clauses: $\{\bigvee_{x_i \in C} x_i : C \in \gamma(G)\}$
  - Computes a coloring $\gamma(G)$.
  - One soft clause for each color.



Hard clauses: $\{(\neg 1 \vee \neg 3), (\neg 1 \vee \neg 4), (\neg 1 \vee \neg 5), (\neg 3 \vee \neg 4), (\neg 4 \vee \neg 5)\}$
Soft clauses: $\{(1 \vee 4), (2), (3), (5)\}$

This Work

- Using both reductions, experimentally compare current PMaxSat solvers and branch and bound algorithms for MC.
- Identify parallel between the decision making of PMaxSat solvers and branching for MC.

Experiments

- Dedicated algorithm: MCQD (Konc and Janežič, 2007) (available at http://insilab.org/maxclique/)
- PMaxSat solvers:
  - MiniMaxSat (Heras, Larrosa, and Oliveras, 2008)
  - AhMaxSat (Abrame and Habet, 2015) (available at http://www.lsis.org/habetd/Djamal_Habet/MaxSAT.html)
- Instances from the Second DIMACS Implementation Challenge and random graphs in the GNP model.

Experimental Results – DIMACS

	MCQD		MiniMaxSat		MiniMaxSat (color)	
Instance	**Tree size**	**Time** (s)	**Tree size**	**Time** (s)	**Tree size**	**Time** (s)
MANN_a9	99	0.0002	390	0.00	182	0.02
brock200_1	235172	0.8043	2783359	17.75	2794161	16.89
brock200_2	3622	0.0099	58747	0.41	67982	0.44
brock200_3	12706	0.0470	196971	1.41	169383	1.16
brock200_4	46755	0.1346	581646	3.78	456447	2.86
c-fat200-1	212	0.0005	402	0.71	229	0.05
c-fat200-2	238	0.0006	407	0.42	325	0.05
c-fat200-5	307	0.0013	391	0.04	391	0.05
c-fat500-1	513	0.0021	1024	0.38	267	0.40
c-fat500-5	618	0.0047	969	0.35	415	0.32
hamming6-2	62	0.0002	65	0.00	33	0.03
hamming6-4	123	0.0001	826	0.69	629	0.03
hamming8-2	371	0.0088	613	0.03	129	0.40
hamming8-4	12704	0.0543	2207385	14.06	1065844	6.40
hamming10-2	2863	2.5956	8423	0.42	513	0.06
johnson8-2-4	50	0.0001	182	0.02	213	0.02
johnson8-4-4	216	0.0006	2920	0.03	3902	0.02
johnson16-2-4	583334	0.5403	5021662	10.72	8733582	16.28
keller4	7711	0.0198	763074	2.90	279042	2.76
p_hat300-1	2129	0.0074	22576	0.30	18631	0.24
p_hat300-2	7459	0.0307	77678	0.96	55125	0.73
p_hat300-3	629203	3.8099	5143557	50.28	4106223	39.45
p_hat500-1	10956	0.0290	167942	3.67	154397	3.15
p_hat500-2	193480	1.2548	1036800	27.25	887731	22.60

Experimental Results – DIMACS

	MCQD		**MiniMaxSat**		**MiniMaxSat** (color)	
Instance	**Tree size**	**Time** (*s*)	**Tree size**	**Time** (*s*)	**Tree size**	**Time** (*s*)
p_hat300-1	2129	0.0074	22576	0.30	18631	0.24
p_hat300-2	7459	0.0307	77678	0.96	55125	0.73
p_hat300-3	629203	3.8099	5143557	50.28	4106223	39.45
p_hat500-1	10956	0.0290	167942	3.67	154397	3.15
p_hat500-2	193480	1.2548	1036800	27.25	887731	22.60
san200_0.7_1	1138	0.0058	529637137	1572.80	410473000	1174.73
san200_0.9_1	11031	0.0798	84871	0.91	240361	1.43
san200_0.9_2	49351	0.4439	2400718	22.81	671891	5.97
san200_0.9_3	332964	2.6274	9116566	104.40	482754	5.81
san400_0.9_1	628188	19.9002	601647032	2781.66	84159929	713.46
sanr200_0.7	106744	0.3240	1269259	12.05	1198260	11.71
sanr200_0.9	6361069	41.0665	49717078	491.60	41300741	376.94
sanr400_0.5	242343	0.6782	4100323	77.55	3902503	61.51
SUM	9483731	74.48	1216050993	5200.36	561225215	2465.97
Ratio of McqDYN			128.22		59.18	

Experimental Results – DIMACS

	MCQD		AhMaxSat		AhMaxSat (color)	
Instance	Tree size	Time (s)	Tree size	Time (s)	Tree size	Time (s)
MANN_a9	99	0.0002	93	0.01	92	0.00
brock200_1	235172	0.8043	145224	58.75	122104	46.06
brock200_2	3622	0.0099	9506	2.85	7216	2.02
brock200_3	12706	0.0470	24190	7.42	25025	6.82
brock200_4	46755	0.1346	59145	18.27	29683	9.55
c-fat200-1	212	0.0005	203	1.13	209	0.35
c-fat200-2	238	0.0006	214	0.92	220	0.33
c-fat200-5	307	0.0013	198	0.43	143	0.30
c-fat500-1	513	0.0021	559	16.94	624	45.27
c-fat500-5	618	0.0047	515	13.02	561	9.18
hamming6-2	62	0.0002	36	0.00	36	0.03
hamming6-4	123	0.0001	174	0.70	133	0.03
hamming8-2	371	0.0088	203	0.49	180	0.83
hamming8-4	12704	0.0543	65269	26.64	29698	12.74
hamming10-2	2863	2.5956	1188	61.11	1287	61.61
johnson8-2-4	50	0.0001	49	0.00	39	0.00
johnson8-4-4	216	0.0006	162	0.03	132	0.01
johnson16-2-4	583334	0.5403	400506	16.70	206492	6.59
keller4	7711	0.0198	34695	5.24	13536	3.51
p_hat300-1	2129	0.0074	5169	7.23	3185	5.91
p_hat300-2	7459	0.0307	17797	21.27	5700	6.16
p_hat300-3	629203	3.8099	343461	467.91	508717	580.55
p_hat500-1	10956	0.0290	36577	58.47	23753	39.74
p_hat500-2	193480	1.2548	496326	896.25	153383	224.98

Experimental Results – DIMACS

	MCQD		AhMaxSat		AhMaxSat (color)	
Instance	**Tree size**	**Time (s)**	**Tree size**	**Time (s)**	**Tree size**	**Time (s)**
san200_0.7_1	1138	0.0058	2193	2.06	1224	1.15
san200_0.9_1	11031	0.0798	383	0.25	419	0.17
san200_0.9_2	49351	0.4439	307	0.41	1042	1.48
san200_0.9_3	332964	2.6274	62251	83.24	13231	17.35
san400_0.9_1	628188	19.9002	327389	909.62	65113	209.90
sanr200_0.7	106744	0.3240	77541	47.00	59213	34.48
sanr200_0.9	6361069	41.0665	143283	232.76	138638	198.53
sanr400_0.5	242343	0.6782	373764	274.76	346190	256.11
SUM	9483731	74.48	2628570	3231.88	1757218	1781.74
Ratio of McqDYN			0.28		0.19	

Conclusions

- The time using a PMaxSat solver is really not competitive.
- Trees generated by AhMaxSat are smaller than by the dedicate algorithm.
- Experiments in random graphs (not shown here) showed similar results, except that MiniMaxSat presented better times than AhMaxSat.
- There is still some potential for improvements in dedicated algorithms by adapting PMaxSat techniques.
- One particular difference: the PMaxSat solver might consider the negation of a variable first, whereas all dedicated algorithms always insert a vertex in the clique first.

References

Andre Abrame and Djamal Habet. Ahmaxsat: Description and evaluation of a branch and bound max-sat solver. *Journal on Satisfiability, Boolean Modeling and Computation*, 9:89–128, 2015.

Federico Heras, Javier Larrosa, and Albert Oliveras. Minimaxsat: An efficient weighted max-sat solver. *J. Artif. Intell. Res.(JAIR)*, 31:1–32, 2008.

Janez Konc and Dušanka Janežič. An improved branch and bound algorithm for the maximum clique problem. *MATCH Communications in Mathematical and in Computer Chemistry*, June 2007. URL http://www.sicmm.org/konc/articles/match2007.pdf.

Chu-Min Li and Zhe Quan. An efficient branch-and-bound algorithm based on MaxSAT for the maximum clique problem. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, pages 128–133, 2010a. URL http://www.aaai.org/ocs/index.php/AAAI/AAAI10/paper/view/1611.

Chu-Min Li and Zhe Quan. Combining graph structure exploitation and propositional reasoning for the maximum clique problem. In *2010 22nd IEEE International Conference on Tools with Artificial Intelligence*, volume 1, pages 344–351. IEEE, 2010b. doi: 10.1109/ictai.2010.57. URL http://dx.doi.org/10.1109/ICTAI.2010.57.

Chu-Min Li, Zhiwen Fang, and Ke Xu. Combining MaxSAT reasoning and incremental upper bound for the maximum clique problem. In *ICTAI*, pages 939–946. IEEE, 2013. URL http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=6734837.

Chu-Min Li, Hua Jiang, and Ru-Chu Xu. Incremental maxsat reasoning to reduce branches in a branch-and-bound algorithm for maxclique. In *International Conference on Learning and Intelligent Optimization*, pages 268–274. Springer, 2015.

Pablo San Segundo, Alexey Nikolaev, and Mikhail Batsyn. Infra-chromatic bound for exact maximum clique search. *Computers & Operations Research*, 64:293–303, 2015. ISSN 0305-0548. doi: http://dx.doi.org/10.1016/j.cor.2015.06.009. URL http://www.sciencedirect.com/science/article/pii/S0305054815001549.

Thank you!

Alexandre Prusch Züge
alexandrezuge@ufpr.br