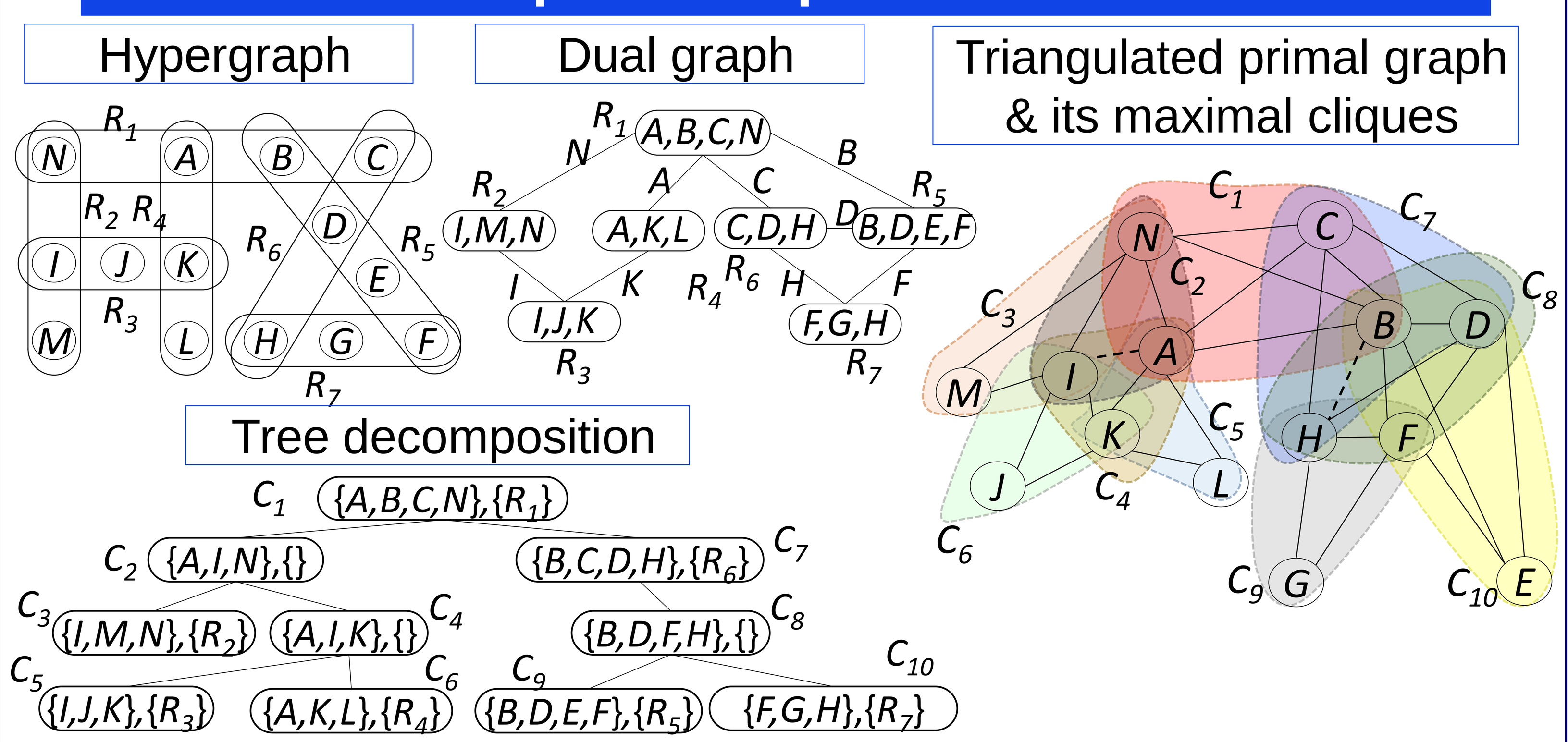
Improving the Performance of Consistency Algorithms by Localizing and Bolstering Propagation in a Tree Decomposition

Shant Karakashian, Robert J. Woodward and Berthe Y. Choueiry
Constraint Systems Laboratory, University of Nebraska-Lincoln, USA

Contributions

1. Localizing consistency to the clusters of a tree decomposition of a CSP & bolstering propagation at the separators between clusters
2. Theoretical characterization of resulting new consistency properties
3. Empirical evaluation of our approach establishing its benefits on difficult benchmarks, solving many problems in a backtrack-free manner and, thus, approaching 'practical tractability'

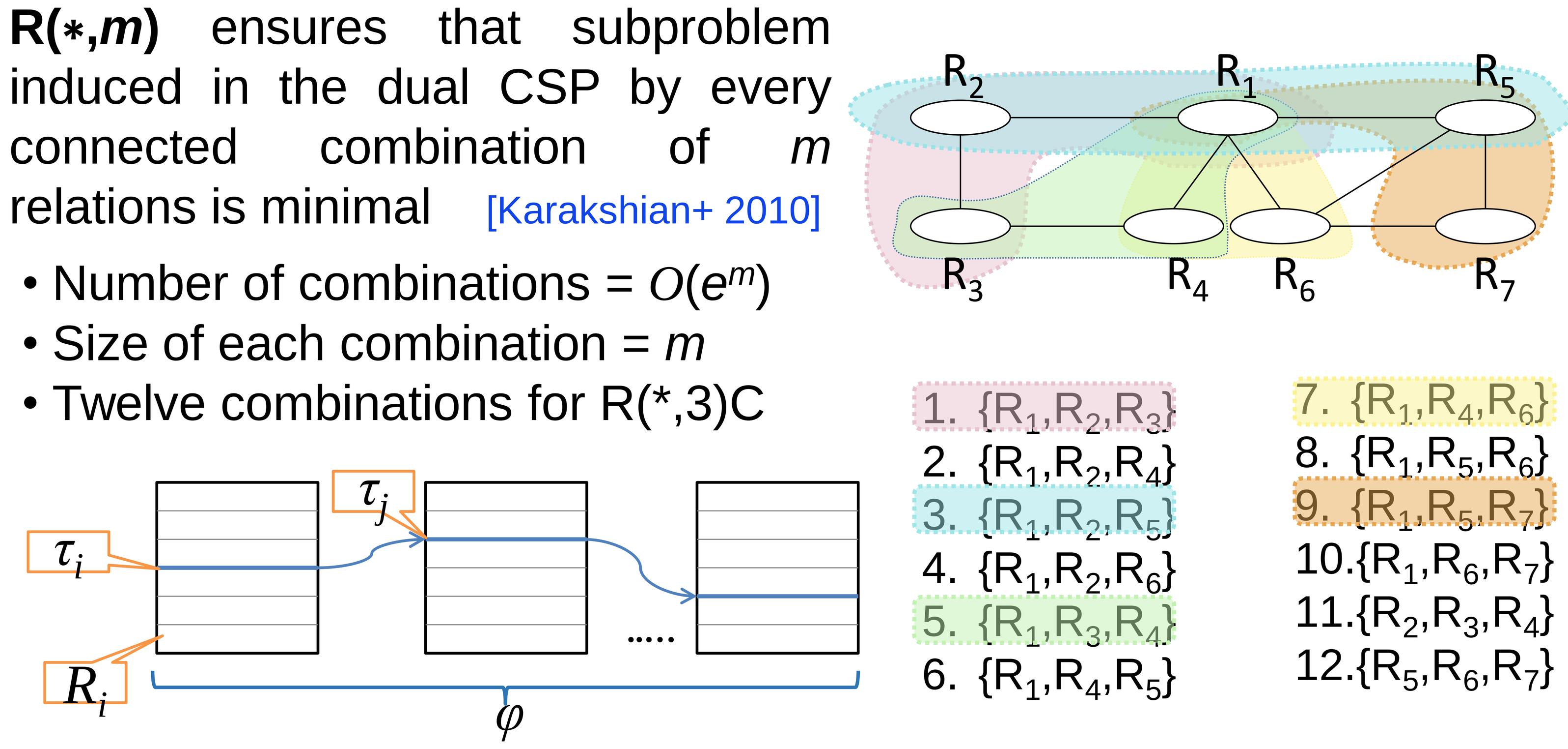
Graphical Representation



Relational Consistency

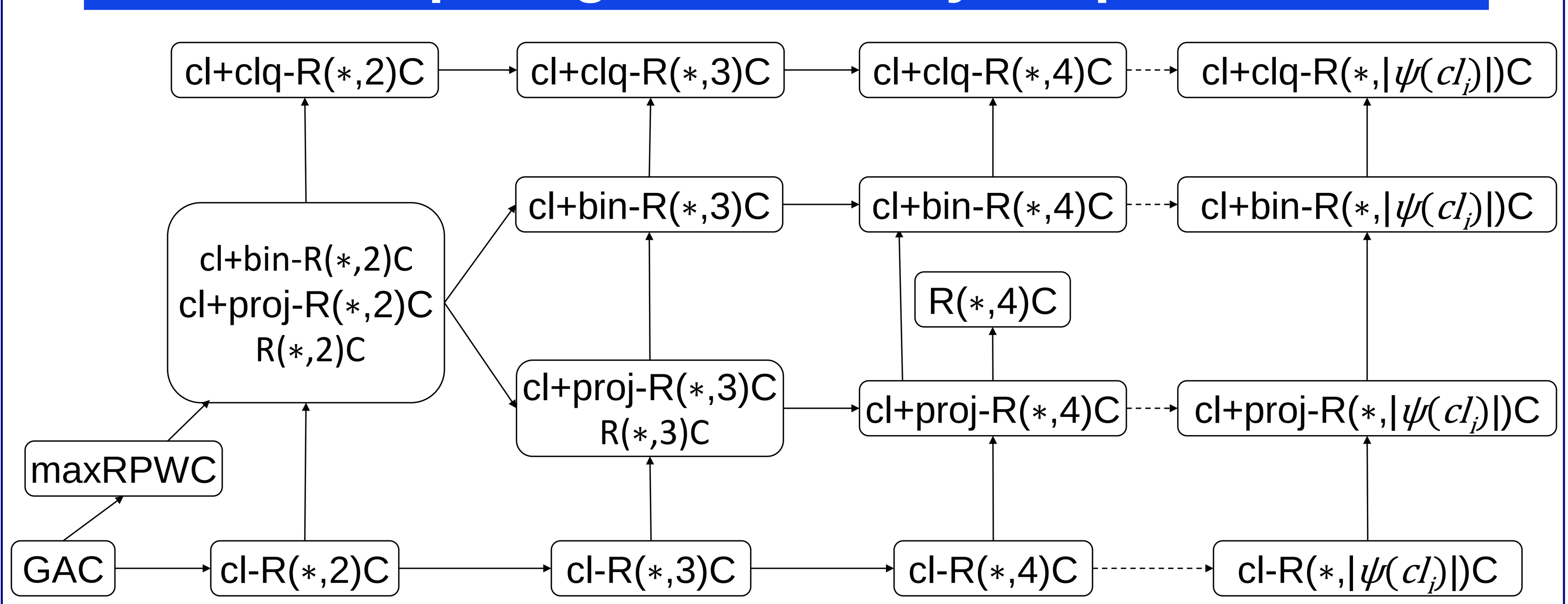
R(∗,*m*) ensures that subproblem induced in the dual CSP by every connected combination of m relations is minimal [Karakashian+ 2010]

- Number of combinations = $O(e^m)$
- Size of each combination = m
- Twelve combinations for R(*,3)C



1. $\{R_1,R_2,R_3\}$
2. $\{R_1,R_2,R_4\}$
3. $\{R_1,R_2,R_5\}$
4. $\{R_1,R_2,R_6\}$
5. $\{R_1,R_3,R_4\}$
6. $\{R_1,R_4,R_5\}$
7. $\{R_1,R_4,R_6\}$
8. $\{R_1,R_5,R_6\}$
9. $\{R_1,R_5,R_7\}$
10. $\{R_1,R_6,R_7\}$
11. $\{R_2,R_3,R_4\}$
12. $\{R_5,R_6,R_7\}$

Comparing Consistency Properties



Bolstering Propagation at Separators

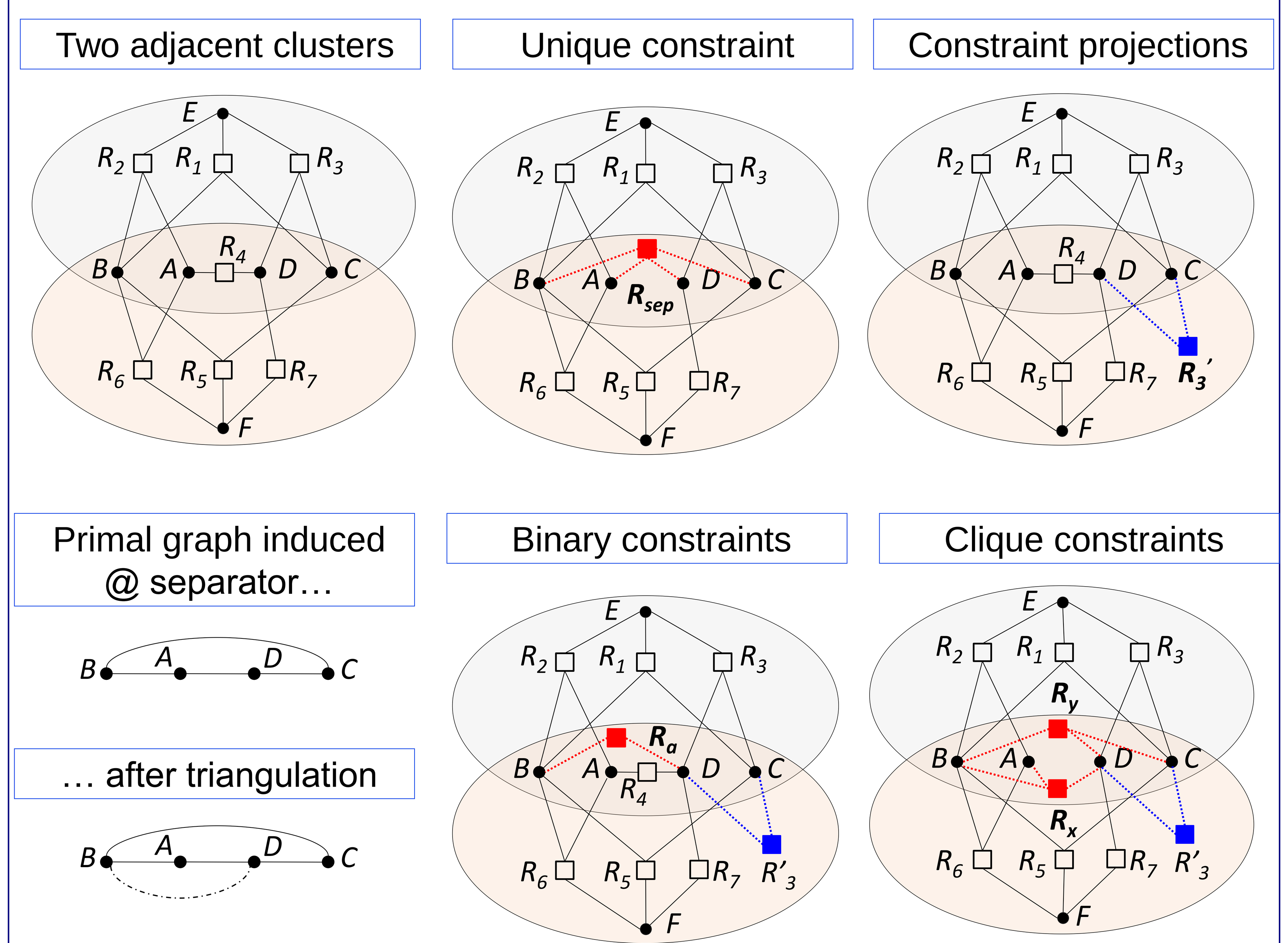
Localizing R(∗,*m*)C

We localize the application of the consistency algorithm to each cluster of the tree decomposition, which allows us to increase the value of m to the number of relations in the cluster and, thus, the level of consistency enforced.

Bolstering Separators

We bolster constraint propagation along the tree by adding redundant constraints to the separators.

- A perfect 'communication' between clusters requires a unique constraint over the separator's variables, but materializing such a constraint is prohibitive in terms of space [Fattah and Dechter 1996; Kask+ 2005].
- We propose three approximation schemes to this end: constraint projection (proj), binary constraints (bin), clique constraints (clq).



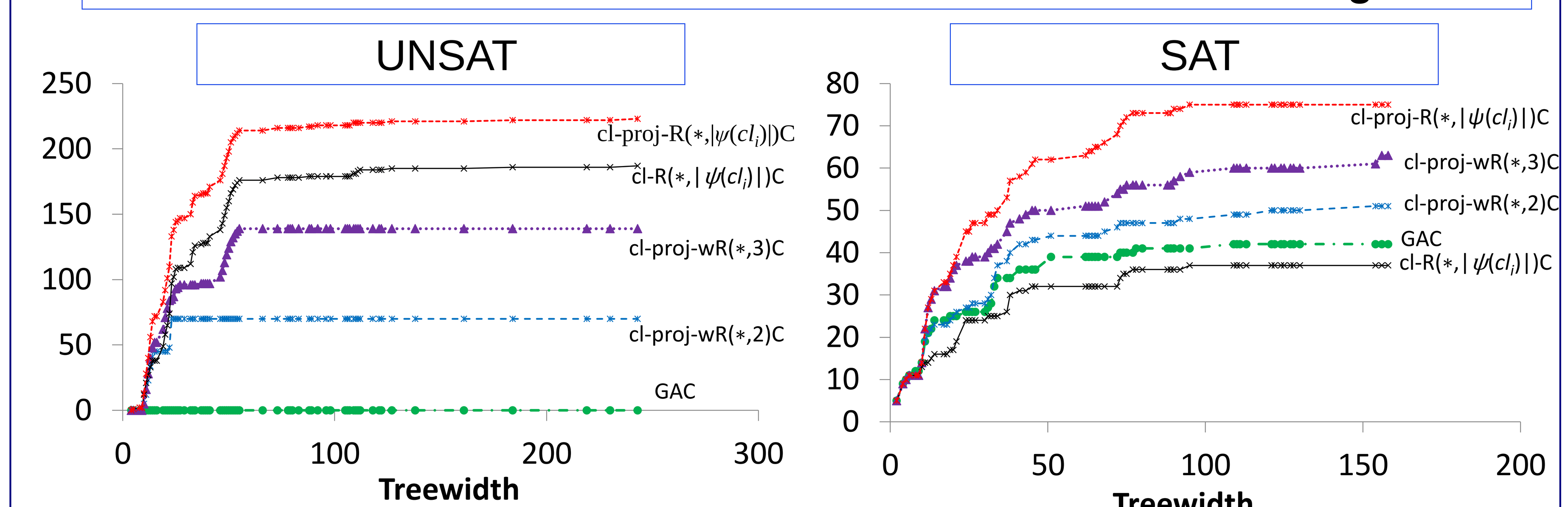
Data Characteristics

	max		median		mean	
	UNSAT	SAT	UNSAT	SAT	UNSAT	SAT
treewidth	243	158	33	18	43.45	34.44
largest sep.	214	157	28	16.5	39.02	31.33
Max(|ψ(cl_i)|)						
local	1,243	211	16	8	109.54	18.35
projection	1,243	211	18	11	114.70	37.35
binary	1,243	653	24	12	199.50	80.65
clique	1,243	148	18	10	113.35	25.87
clique arity	48	26	7	4	7.40	5.97

Empirical Evaluation

+ maxRPWC, m=3,4	#inst.	GAC	wR(∗,2)C global	wR(∗,2)C local	wR(∗,2)C Proj.	wR(∗,2)C binary	wR(∗,2)C clique	R(∗,|ψ(cl_i)|)C local	R(∗,|ψ(cl_i)|)C Proj.	R(∗,|ψ(cl_i)|)C binary	R(∗,|ψ(cl_i)|)C clique
Completed	UNSAT	167	170	167	172	169	162	285	**286**	282	271
	479	34.9%	35.5%	34.9%	35.9%	35.3%	33.8%	59.5%	**59.7%**	58.9%	56.6%
	SAT	174	**179**	178	176	169	104	152	138	124	113
	200	87.0%	**89.5%**	89.0%	88.0%	84.5%	52.0%	76.0%	69.0%	62.0%	56.5%
BT-Free	UNSAT	0	70	39	70	70	74	187	**223**	223	213
	479	0.0%	14.6%	8.1%	14.6%	14.6%	15.4%	39.0%	**46.6%**	46.6%	44.5%
	SAT	44	55	37	53	52	38	39	**77**	71	58
	200	22.0%	27.5%	18.5%	26.5%	26.0%	19.0%	19.5%	**38.5%**	35.5%	29.0%
Min(#NV)	UNSAT	17	73	43	72	72	77	220	**249**	248	239
	479	3.5%	15.2%	9.0%	15.0%	15.0%	16.1%	45.9%	**52.0%**	51.8%	49.9%
	SAT	47	64	37	62	61	39	83	**111**	100	79
	200	23.5%	32.0%	18.5%	31.0%	30.5%	19.5%	41.5%	**55.5%**	50.0%	39.5%
Fastest	UNSAT	72	13	35	5	1	1	**176**	108	42	37
	479	15.0%	2.7%	7.3%	1.0%	0.2%	0.2%	**36.7%**	22.5%	8.8%	7.7%
	SAT	**121**	45	47	23	14	12	34	18	13	12
	200	**60.5%**	22.5%	23.5%	11.5%	7.0%	6.0%	17.0%	9.0%	6.5%	6.0%

Cumulative count of instances solved w/o backtracking



- Localization & projection always outperformed 'global'
- Bolstering schemes of increasing 'sophistication' investigate tradeoff between
  - the effectiveness of constraint filtering &
  - the cost of generating and maintaining redundant constraints
- In tests, projection yields the right tradeoff between the strength of the enforced consistency & the overhead of processing redundant constraints

Future Research Directions

- Automatic selection of consistency property: inside clusters & during search
- Modify the structure of a tree decomposition to improve performance (e.g., merging clusters to reduce the overlap [Fattah and Dechter 1996])

Experiments were conducted on the equipment of the Holland Computing Center at the University of Nebraska-Lincoln.
This research is supported by NSF Grant No. RI-111795.

July 17th, 2013