A New Solver for the Minimum Weighted Vertex Cover Problem

Hong Xu, T. K. Satish Kumar, Sven Koenig

USC University of Southern California

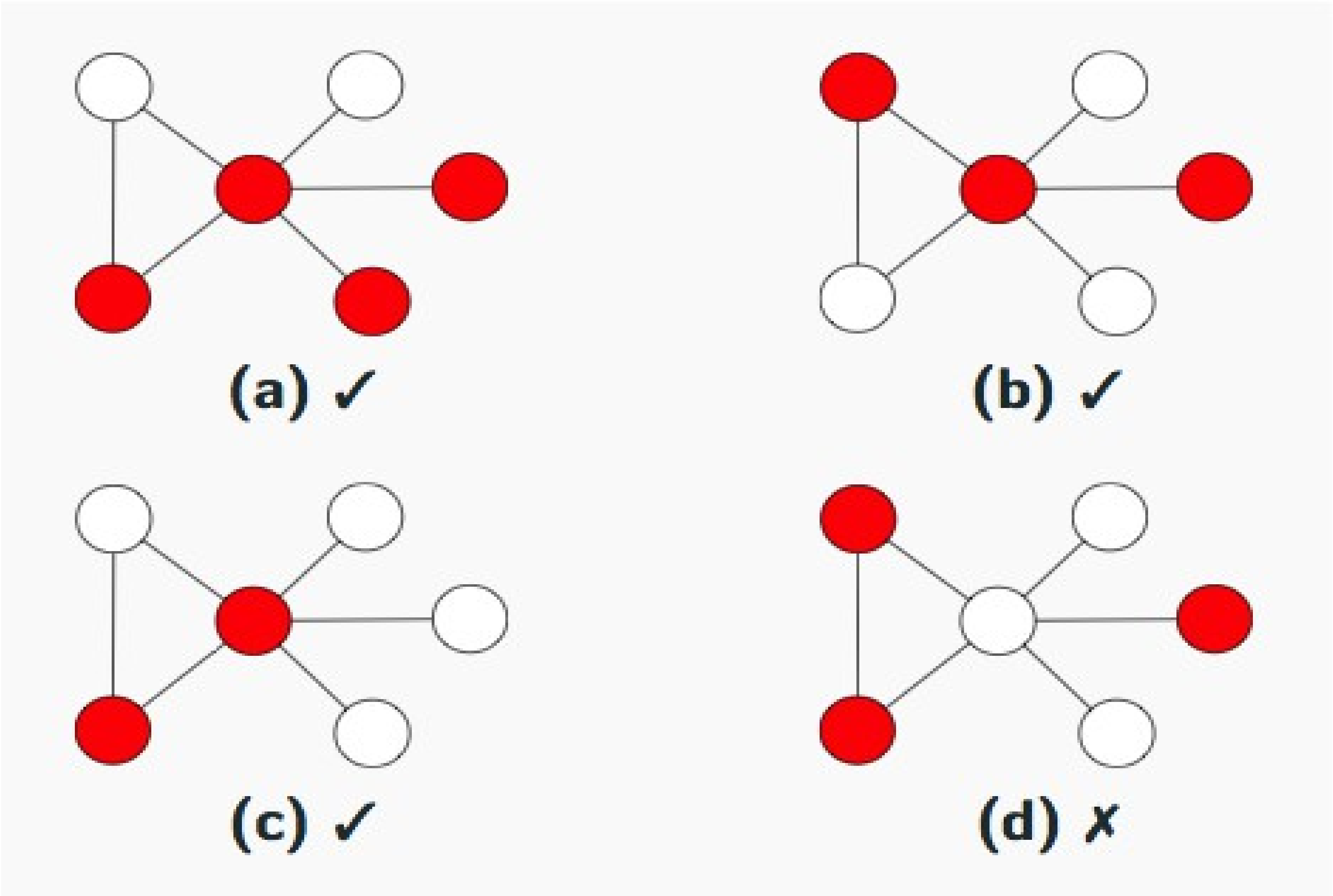
Roadmap

- What is the ***minimum weighted vertex cover*** (MWVC) problem?
- Why is it so important?
  - weighted constraint satisfaction problems
  - constraint composite graphs
- How do we solve it efficiently?
  - previous approaches
  - proposed method
- Conclusions and future work

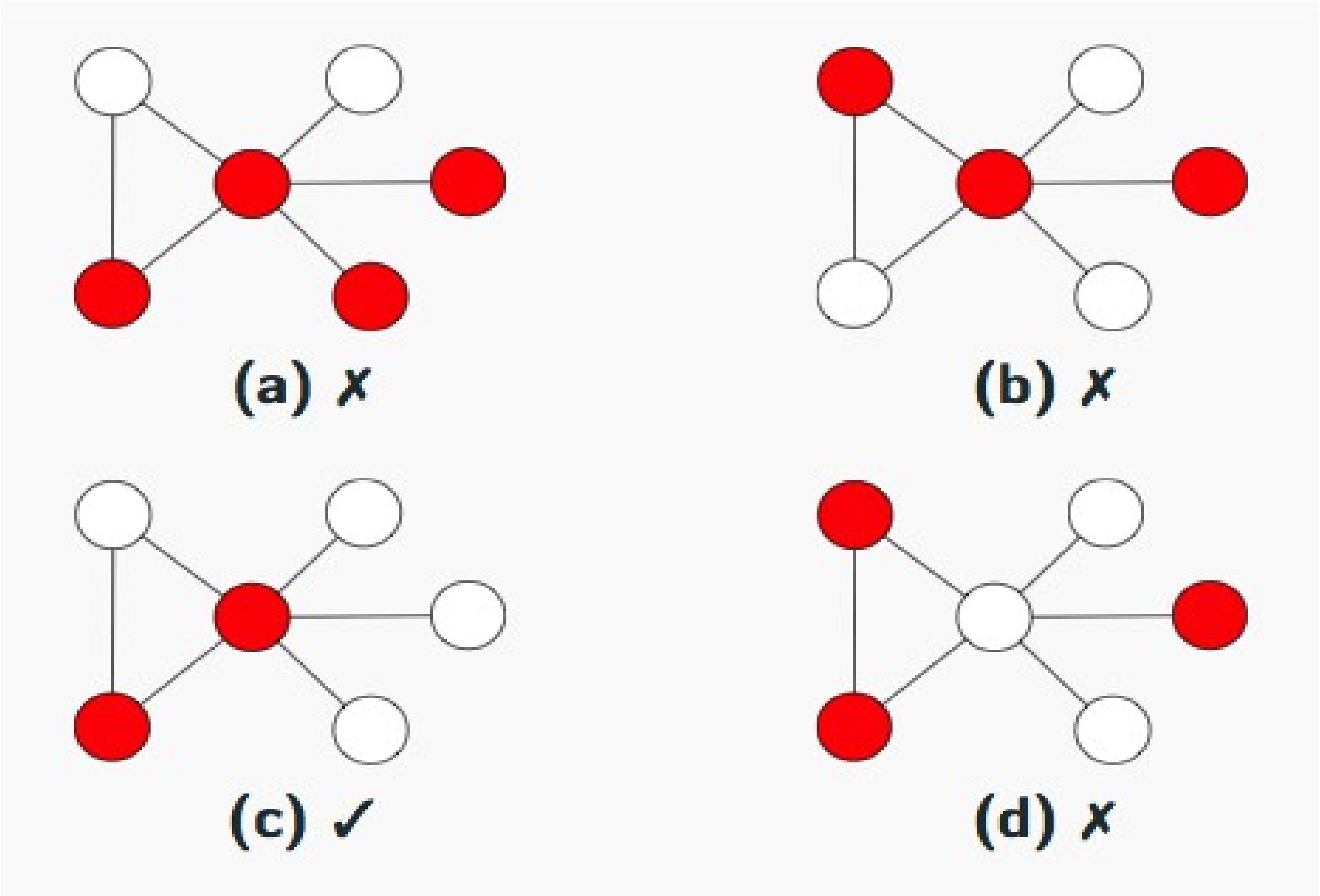
Roadmap

- What is the ***minimum weighted vertex cover*** (MWVC) problem?
- Why is it so important?
  - weighted constraint satisfaction problems
  - constraint composite graphs
- How do we solve it efficiently?
  - previous approaches
  - proposed method
- Conclusions and future work

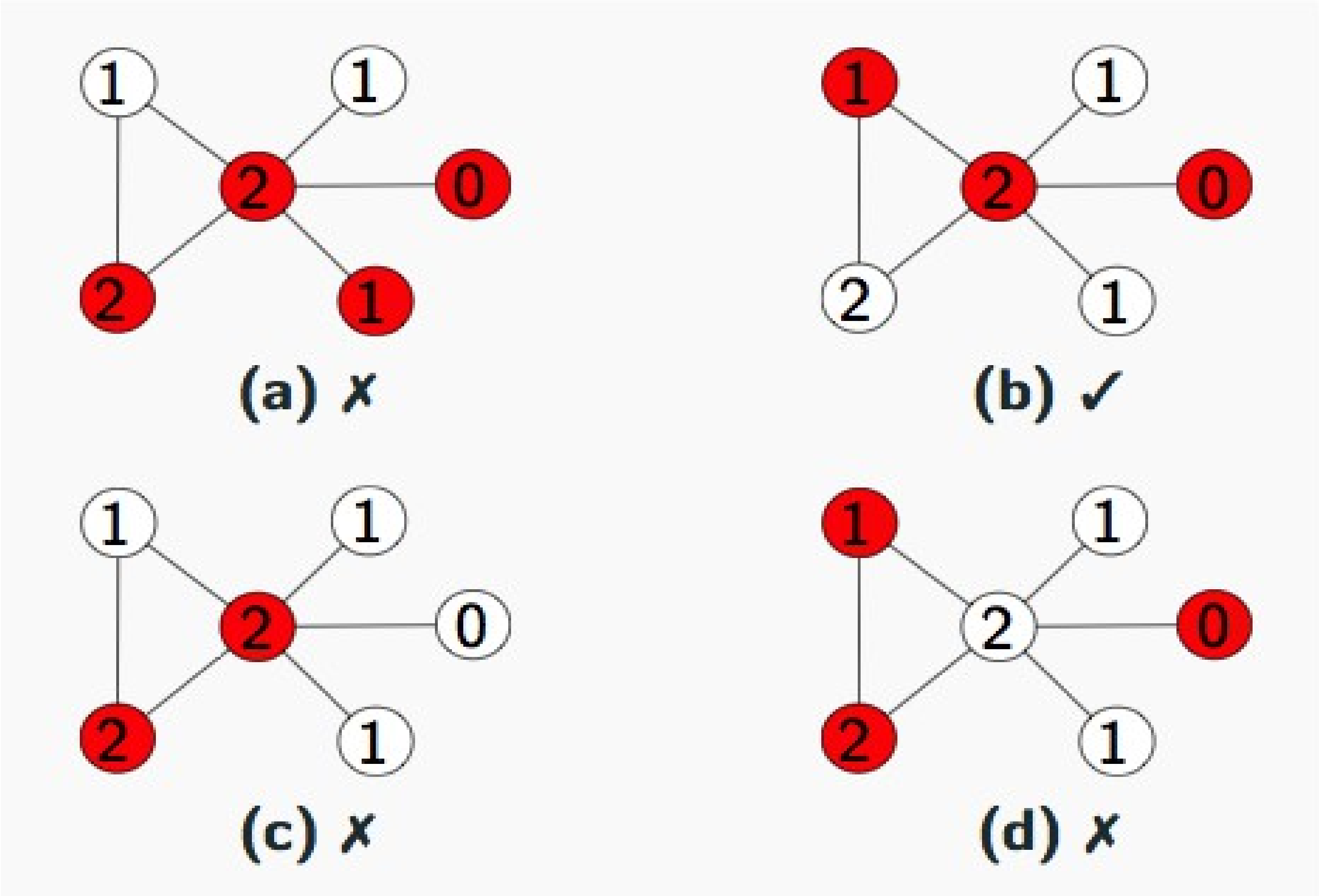
Vertex Cover



Minimum Vertex Cover



Minimum Weighted Vertex Cover



Complexity Results

- Both the MVC problem and the MWVC problem are NP-hard to solve optimally.

- But both problems are amenable to a polynomial-time factor-2 approximation algorithm.

- The MVC problem is fixed-parameter tractable; but the MWVC problem is not.

Roadmap

- What is the ***minimum weighted vertex cover*** (MWVC) problem?
- Why is it so important?
  - weighted constraint satisfaction problems
  - constraint composite graphs
- How do we solve it efficiently?
  - previous approaches
  - proposed method
- Conclusions and future work

Constraint Satisfaction Problems

- A *Constraint Satisfaction Problem* (CSP) is characterized by:

- *N* discrete-valued variables $\{X_1, X_2 \ldots X_N\}$
- Each variable X_i has a discrete domain D_i associated with it, from which it can take values.
- *M* constraints $\{C_1, C_2 \ldots C_M\}$
- Each constraint C_i specifies, for some subset of the variables, the allowed and disallowed combinations of values to them.
- A *solution* is an assignment of values to all variables from their respective domains such that all constraints are satisfied.

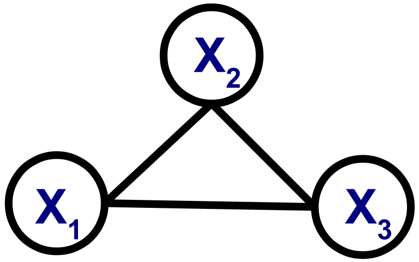
Weighted CSPs

- *N* variables $X_1, X_2 \ldots X_N$
- Each variable X_i has a discrete-valued domain D_i.

- *M weighted* constraints $C_1, C_2 \ldots C_M$
- Each constraint C_i specifies the *cost* for every combination of values to a subset of the variables.

- An *optimal solution* is an assignment of values to all variables from their respective domains so that the *sum* of the costs is *minimized*.

Example Boolean WCSP



X_1	
0	0.8
1	0.2

X_2	
0	0.3
1	0.7

X_3	
0	0.1
1	0.9

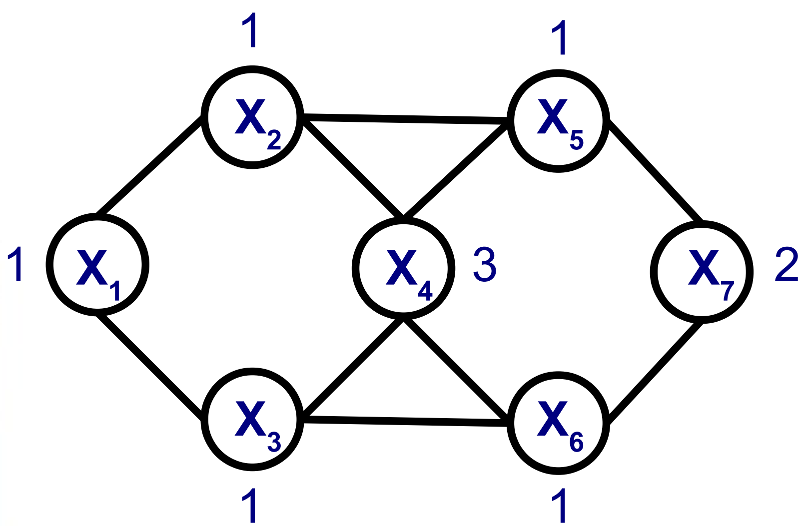
X_1 \ X_2	0	1
0	0.5	0.6
1	0.7	0.3

X_2 \ X_3	0	1
0	0.6	1.3
1	1.0	1.1

X_1 \ X_3	0	1
0	0.4	0.9
1	0.7	0.8

Projections of Minimum Vertex Covers onto Independent Sets

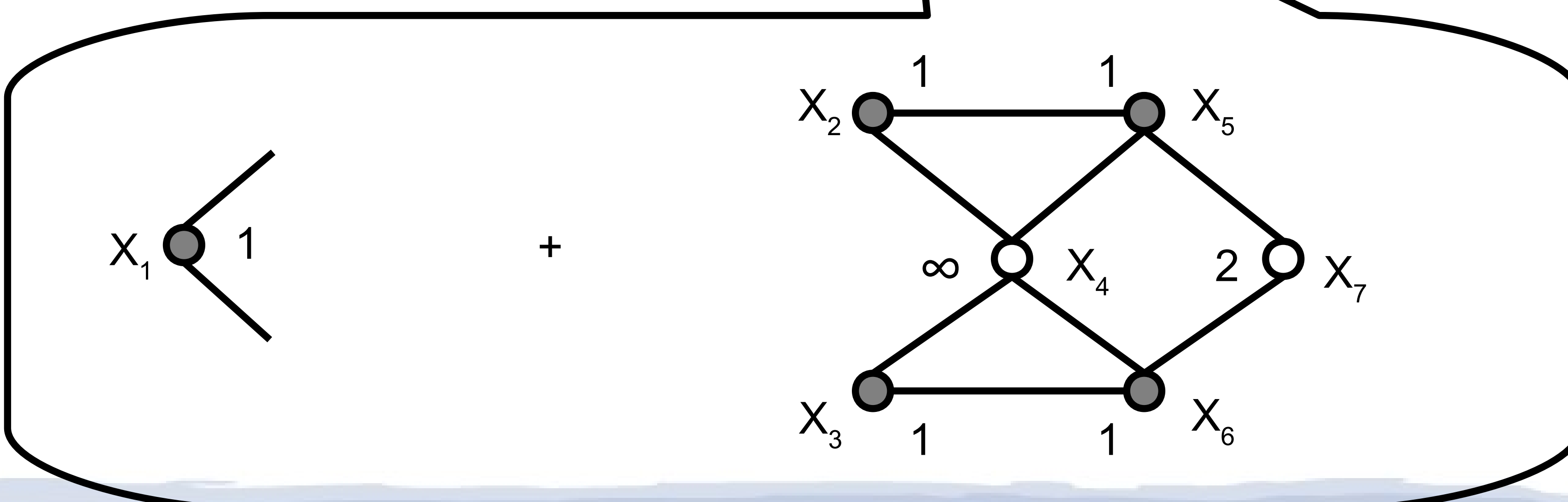
[Kumar, CP2008; Kumar, ISAIM2008]



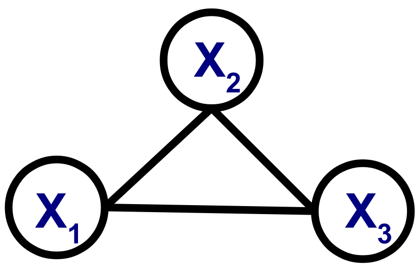
X_1 \ X_4	0	1
0	4	7
1	5	6

1 = necessarily present in the vertex cover

0 = necessarily absent from the vertex cover



Example Boolean WCSP



X_1	
0	0.8
1	0.2

X_2	
0	0.3
1	0.7

X_3	
0	0.1
1	0.9

X_1 \ X_2	0	1
0	0.5	0.6
1	0.7	0.3

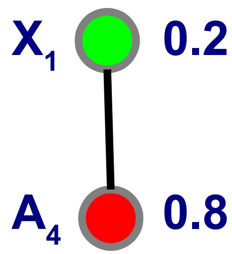
X_2 \ X_3	0	1
0	0.6	1.3
1	1.0	1.1

X_1 \ X_3	0	1
0	0.4	0.9
1	0.7	0.8

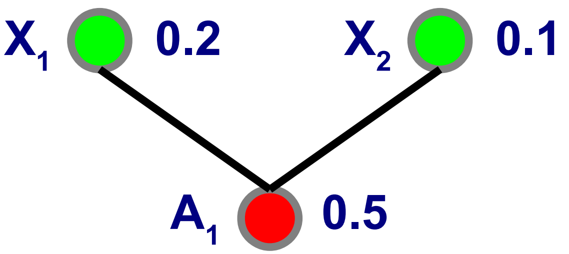
Lifted Representations for Each Weighted Constraint

[Kumar, CP2008; Kumar, ISAIM2008]

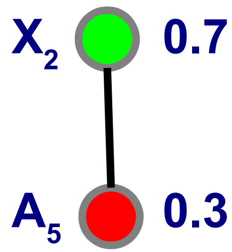
X_1	
0	0.8
1	0.2



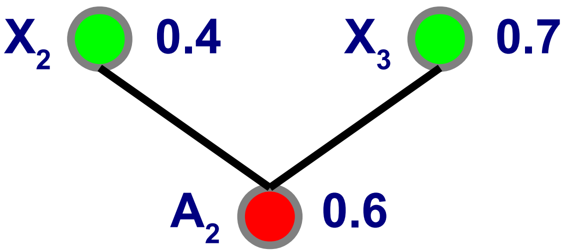
X_1 \ X_2	0	1
0	0.5	0.6
1	0.7	0.3



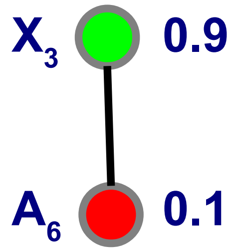
X_2	
0	0.3
1	0.7



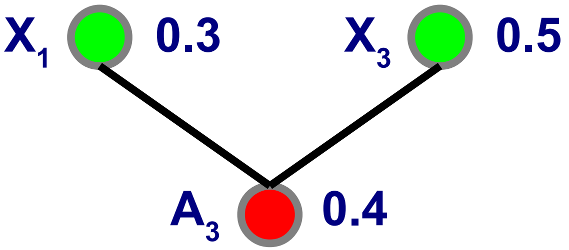
X_2 \ X_3	0	1
0	0.6	1.3
1	1.0	1.1



X_3	
0	0.1
1	0.9

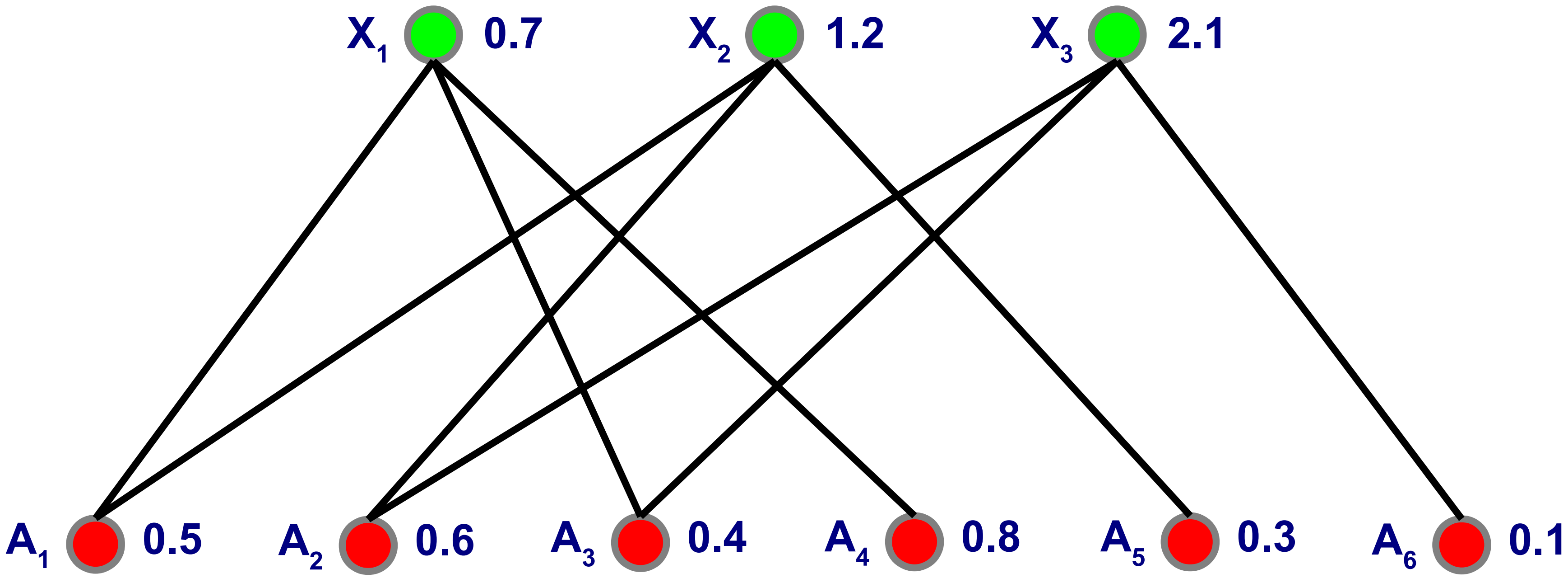


X_1 \ X_3	0	1
0	0.4	0.9
1	0.7	0.8



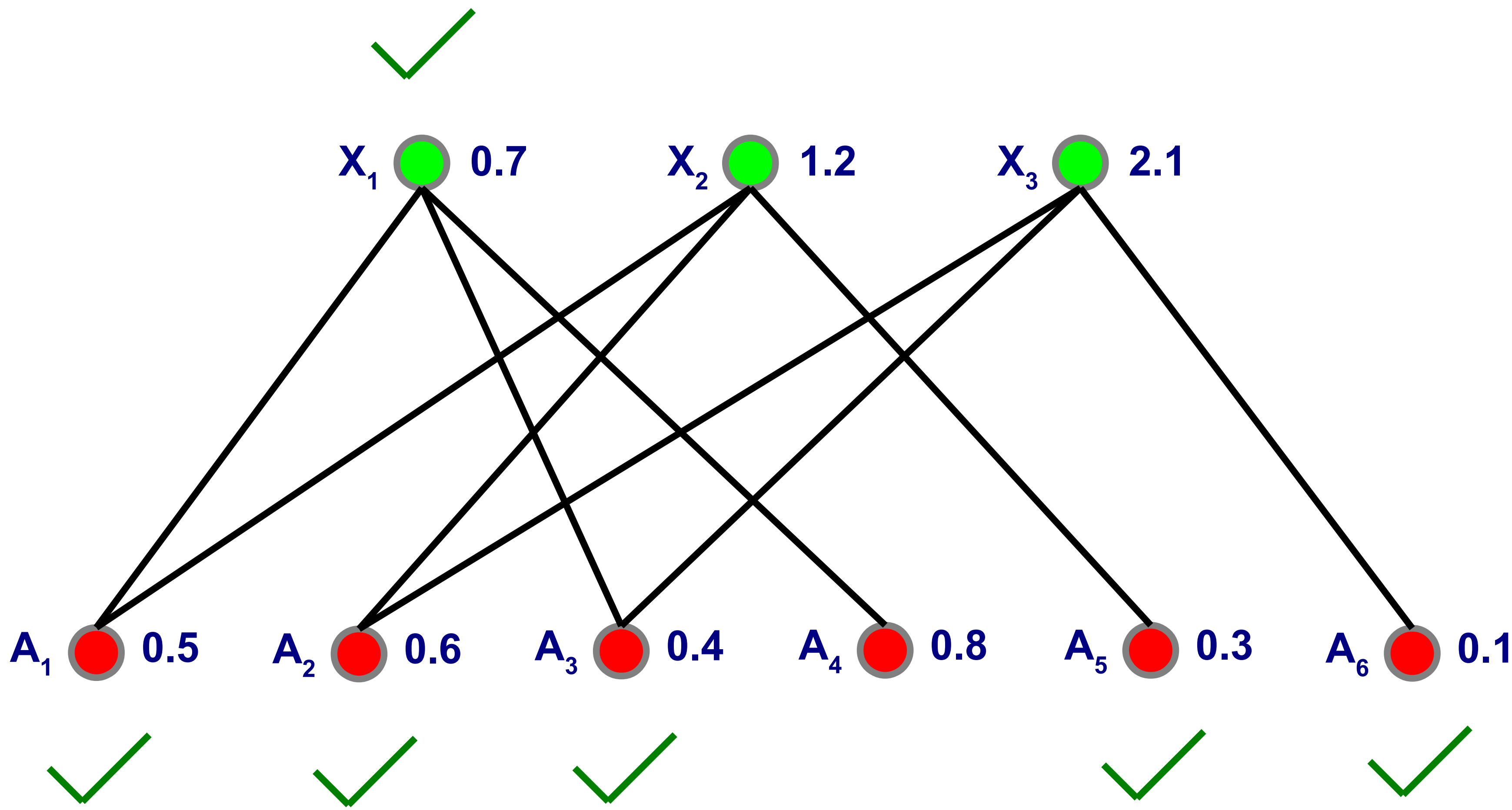
The Constraint Composite Graph

[Kumar, CP2008; Kumar, ISAIM2008]



The Constraint Composite Graph

[Kumar, CP2008; Kumar, ISAIM2008]



A minimum weighted vertex cover of the CCG encodes an optimal solution to the original WCSP!

Roadmap

- What is the ***minimum weighted vertex cover*** (MWVC) problem?
- Why is it so important?
  - weighted constraint satisfaction problems
  - constraint composite graphs
- How do we solve it efficiently?
  - previous approaches
  - proposed method
- Conclusions and future work

Solving the MWVC Problem

- The MVC problem and the MWVC problem are both NP-hard.
- There is a very efficient local search solver for the MVC problem called NuMVC.
- But NuMVC cannot be extended to solve the MWVC problem.
  - The MVC problem is fixed-parameter tractable.
  - This is used critically by NuMVC.

MWVC as an Integer Linear Program

Minimize $\sum_{(i \in V)} w_i X_i$
s.t.
for all $(i,j) \in E$: $X_i + X_j \geq 1$
for all $i \in V$: $X_i \in \{0, 1\}$

Does not work well even with the best ILP solvers like Gurobi.

MWVC as a Pseudo-Boolean Optimization Problem

Minimize $\sum_{(i \in V)} w_i X_i$

s.t.

for all $(i,j) \in E$: $X_i + X_j \geq 1$

for all $i \in V$: $X_i \in \{0, 1\}$

Does not work well even with the best PBO solvers like WBO.

MWVC as an Answer Set Program

$$edge(X,Y) \leftarrow edge(Y,X)$$
$$picked(X) \vee picked(Y) \leftarrow edge(X,Y)$$

Does not work well even with the best ASP solvers like Clingo.

MWVC as Weighted MAX-SAT

- The maximum weighted independent set (MWIS) is the complement of the MWVC.
- The MWIS problem can be encoded as a weighted MAX-SAT problem as follows:
  - for all $i \in V$, add the unit clause X_i with weight w_i
  - for all $(i, j) \in E$, add the binary clause $(X_i' \vee X_j')$ with weight L
  - L is a large weight greater than $\sum_{(i \in V)} w_i$

Does not work well even with the best weighted MAX-SAT solvers like Eva Solver.

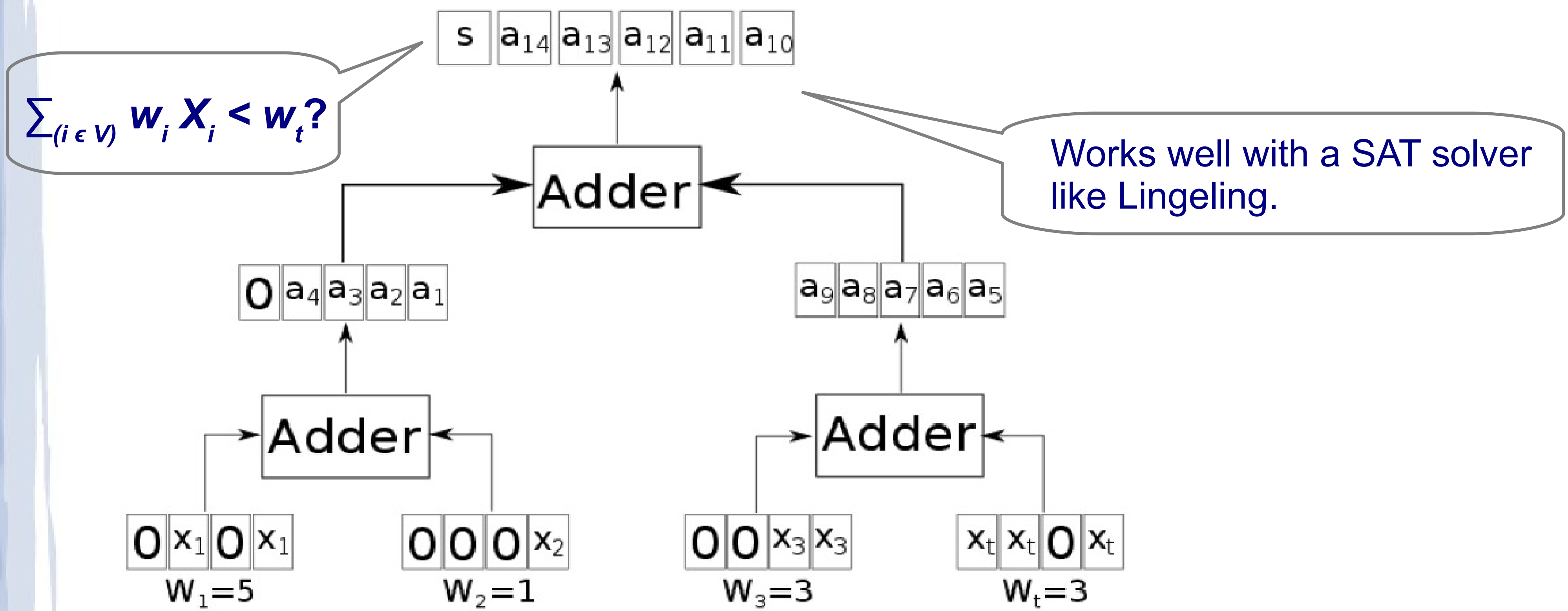
MWVC as Weighted MAX-CLIQUE

- The MWVC problem on a graph is equivalent to the maximum weighted clique problem on its edge-complement graph.

> Does not work well even with the best MAX-CLIQUE solvers like Cliquer.

MWVC as a Series of SAT Instances

- The decision problem "Is there a vertex cover of weight less than a test weight w_t?" can be cast as a SAT problem.



Optimizations in Binary Search

- The MWVC can be found by doing a binary search in the interval $[0, \sum_{(i \in V)} w_i]$.
- We can do much better by starting with the interval $[A/2, A]$. Here, A is the cost of the solution produced by a polynomial-time primal-dual factor-2 approximation algorithm.
- Quasi Binary Search can be used instead of Binary Search.
  - Let current bounds be $[L, U]$ with $w_q = (L+U)/2$.
  - When the Lingeling SAT solver finds a vertex cover of weight $w < w_q$, the bounds for the next iteration can be set to $[L, w]$ instead of $[L, (L+U)/2]$.

Experimental Results

Graph			SBMS				Gurobi		cliquer	
Instance	Vertices	MVC	Running Time	Iteration	Bounds	Initial Bounds	Running Time	Bounds	Running Time	Bounds
frb30-15-1	450	420	49.83	8	-	[218, 437]	22.80	-	15.29	-
frb30-15-2	450	420	40.84	8	-	[219, 438]	11.76	-	30.26	-
frb30-15-3	450	420	36.22	8	-	[218, 437]	34.05	-	120.33	-
frb30-15-4	450	420	40.38	8	-	[219, 439]	29.10	-	0.99	-
frb30-15-5	450	420	34.84	8	-	[219, 438]	10.38	-	0.15	-
frb35-17-1	595	560	65.73	8	-	[292, 584]	84.87	-	14.20	-
frb35-17-2	595	560	84.39	8	-	[292, 584]	>7200	[560, 561]	53.66	-
frb35-17-3	595	560	66.97	8	-	[291, 582]	>7200	[560, 561]	>7200	[-, 582]
frb35-17-4	595	560	55.37	8	-	[292, 584]	>7200	[560, 561]	5189.27	-
frb35-17-5	595	560	54.70	8	-	[290, 581]	>7200	[560, 561]	98.84	-
frb40-19-1	760	720	90.76	8	-	[371, 743]	>7200	[720, 722]	>7200	[-, 736]
frb40-19-2	760	720	131.52	9	-	[372, 745]	>7200	[720, 722]	>7200	[-, 733]
frb40-19-3	760	720	127.73	9	-	[372, 744]	>7200	[720, 721]	273.22	-
frb40-19-4	760	720	243.98	9	-	[372, 744]	>7200	[720, 722]	1555.14	-
frb40-19-5	760	720	198.27	9	-	[372, 745]	>7200	[720, 722]	42.77	-
frb45-21-1	945	900	2955.26	9	-	[465, 930]	>7200	[900, 904]	>7200	[-, 917]
frb45-21-2	945	900	235.59	9	-	[465, 930]	>7200	[900, 903]	>7200	[-, 917]
frb45-21-3	945	900	2036.46	9	-	[465, 930]	>7200	[900, 902]	>7200	[-, 913]
frb45-21-4	945	900	884.90	9	-	[465, 931]	>7200	[900, 902]	>7200	[-, 914]
frb45-21-5	945	900	1958.17	9	-	[465, 931]	>7200	[900, 903]	>7200	[-, 922]
frb50-23-1	1150	1100	3208.50	10	-	[556, 1133]	>7200	[1100, 1104]	>7200	[-, 1102]
frb50-23-2	1150	1100	>7200	9	[1100, 1101]	[567, 1135]	>7200	[1100, 1103]	>7200	[-, 1113]
frb50-23-3	1150	1100	111.09	10	-	[567, 1135]	>7200	[1100, 1105]	>7200	[-, 1112]
frb50-23-4	1150	1100	113.10	10	-	[567, 1135]	>7200	[1100, 1104]	1868.10	-
frb50-23-5	1150	1100	113.68	10	-	[568, 1137]	>7200	[1100, 1104]	>7200	[-, 1129]
frb53-24-1	1272	1219	>7200	8	[1219, 1221]	[625, 1250]	>7200	[1219, 1225]	>7200	[-, 1232]
frb53-24-2	1272	1219	114.87	10	-	[625, 1251]	>7200	[1219, 1224]	>7200	[-, 1239]
frb53-24-3	1272	1219	>7200	9	[1219, 1220]	[628, 1256]	>7200	[1219, 1224]	>7200	[-, 1237]
frb53-24-4	1272	1219	>7200	9	[1219, 1220]	[628, 1257]	>7200	[1219, 1224]	>7200	[-, 1228]
frb53-24-5	1272	1219	120.37	10	-	[627, 1255]	>7200	[1219, 1226]	>7200	[-, 1247]
frb56-25-1	1400	1344	>7200	9	[1344, 1345]	[692, 1384]	>7200	[1344, 1350]	>7200	[-, 1365]
frb56-25-2	1400	1344	>7200	9	[1344, 1345]	[691, 1383]	>7200	[1344, 1352]	>7200	[-, 1371]
frb56-25-3	1400	1344	6717.57	10	-	[692, 1384]	>7200	[1344, 1348]	>7200	[-, 1377]
frb56-25-4	1400	1344	>7200	9	[1344, 1345]	[692, 1385]	>7200	[1344, 1350]	>7200	[-, 1348]
frb56-25-5	1400	1344	120.31	10	-	[690, 1381]	>7200	[1344, 1350]	>7200	[-, 1379]
frb59-26-1	1534	1475	>7200	9	[1475, 1476]	[757, 1514]	>7200	[1475, 1482]	>7200	[-, 1493]
frb59-26-2	1534	1475	>7200	9	[1475, 1476]	[757, 1515]	>7200	[1475, 1481]	>7200	[-, 1513]
frb59-26-3	1534	1475	>7200	9	[1475, 1476]	[757, 1514]	>7200	[1475, 1482]	>7200	[-, 1509]
frb59-26-4	1534	1475	>7200	8	[1475, 1477]	[756, 1513]	>7200	[1475, 1481]	>7200	[-, 1516]
frb59-26-5	1534	1475	131.04	10	-	[759, 1519]	>7200	[1475, 1481]	>7200	[-, 1496]

Unweighted BHOSLIB Instances

Experimental Results

Graph			Running Time of SBMS (mins)							
Instance	Vertices	MWVC	Q+C+N	C+N	Q+C	Q+N	Q	C	N	None
frb30-15-1	450	825	38.33	38.32	37.68	60.00	35.10	37.49	29.99	35.23
frb30-15-2	450	825	59.97	59.98	58.98	75.12	74.87	59.00	75.00	74.80
frb30-15-3	450	790	0.84	0.84	36.43	0.87	36.84	36.32	0.86	36.73
frb30-15-4	450	825	16.92	16.84	14.47	18.79	18.33	14.39	18.80	18.71
frb30-15-5	450	827	28.28	28.34	47.80	27.73	43.13	47.77	27.75	44.35

Weighted BHOSLIB Instances

Diminishing Returns Property

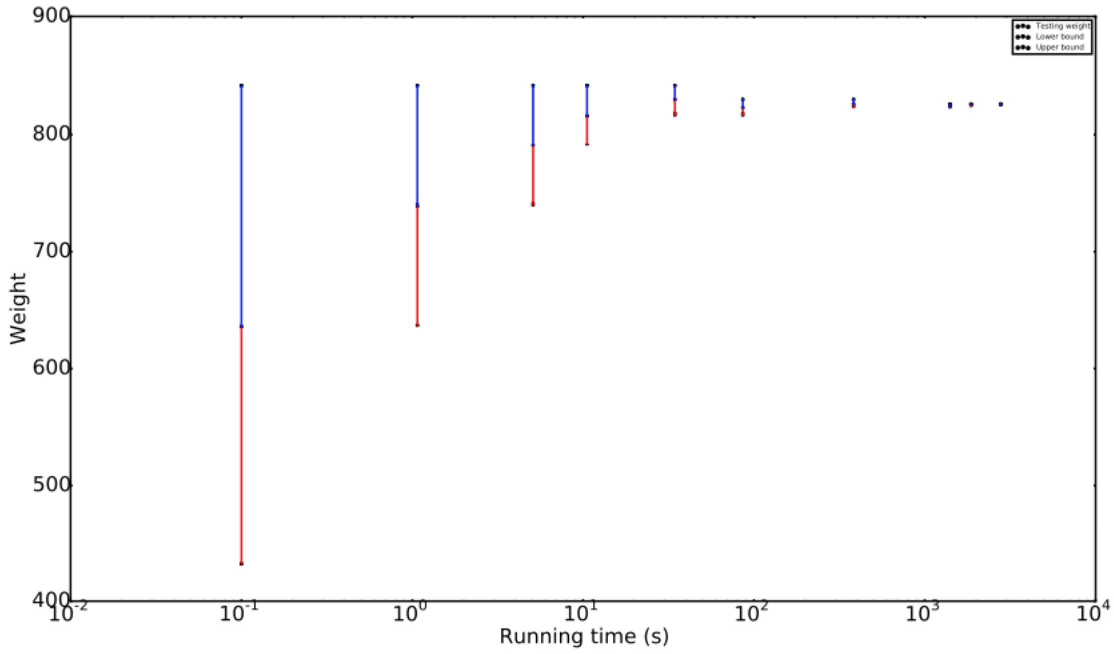


Figure 2: Shows the evolution of the lower and upper bounds with the running time of our SAT-based algorithm on the weighted BHOSLIB instance frb30-15-1. The mid-point of the interval is used as the testing weight for the SAT instance posed at that time.

Roadmap

- What is the ***minimum weighted vertex cover*** (MWVC) problem?
- Why is it so important?
  - weighted constraint satisfaction problems
  - constraint composite graphs
- How do we solve it efficiently?
  - previous approaches
  - proposed method
- Conclusions and future work

Conclusions and Future Work

- The MWVC problem is an important combinatorial problem that can be used to capture the structure in weighted CSPs.
- A feasibility study shows that solving the MWVC problem as a series of SAT instances outperforms other methods.
- In future work, we will use an MWVC solver for efficiently solving weighted CSPs.
  - A new solver for the maximum weighted clique problem published in IJCAI-2016 can be used to our advantage.