Towards Improving Merging Heuristics for Binary Decision Diagrams (BDDs)

LION 13, 2019, May 27-31, Chania, Crete, Greece

Nikolaus Frohner and Günther R. Raidl

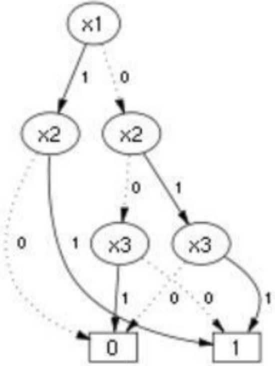
May 30, 2019

ALGORITHMS AND COMPLEXITY GROUP

Binary Decision Diagrams – Basics

- A BDD encodes a function:

x_1	x_2	x_3	f
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1



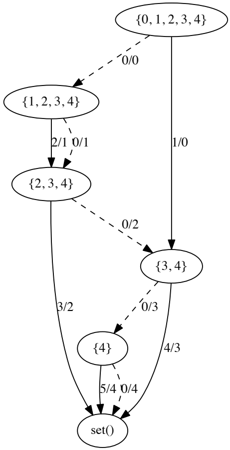
(from Akers (1978))

- Comparable to branching tree, but
- redundant nodes may be removed
- identical subtrees may be superimposed

BDDs for Combinatorial Optimization

Representation of solution space by directed acyclic multigraph with weighted and labeled arcs.

$$X = \{x_1, \ldots, x_n\} \qquad \Leftrightarrow$$



Previous Work

- Lee 1959: BDDs as compact representation for boolean functions.
- Hadzic and Hooker 2006: For post-optimality analysis.
- Bergman et al. 2013: Dual bounds from BDDs for the maximum independent set problem.
- For a compilation of resources: `http://www.andrew.cmu.edu/user/vanhoeve/mdd/`

Paths

- (Partial) solutions for a given problem.
- Carry length via arc costs.
- Define decisions via arc labels.

Longest (shortest) path $\Leftrightarrow$ Maximal (minimal) solution.

Variable Ordering

Decompose solution into parts i and impose an ordering π.

Assign to each subpart π_i of the solution a binary decision variable x_i.

Each subpart can be assigned costs c_{π_i}.

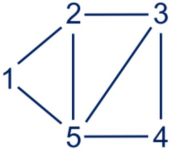
States

Given an i-partial assignment, the corresponding state s_i determines its feasible completions.

States are assigned to nodes in the BDD and determine the completing paths.

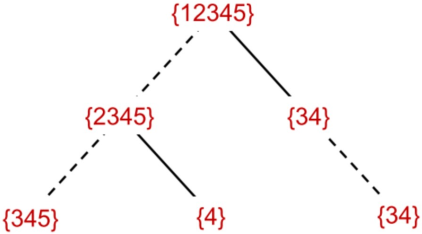
Allows superposition of nodes with same state, resulting into a **reduced BDD**.

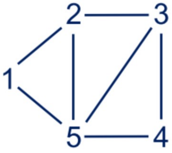
Top-Down Construction of DD



Exact DD for stable set problem

To build DD, associate **state** with each node





x_1

x_2

x_3 Exact DD for stable set problem

x_4

Merge nodes that correspond to the same state

x_5

(from Hooker (2016))

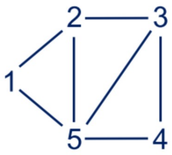
Relaxed BDDs

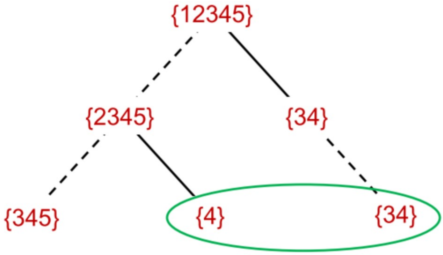
Keep BDDs polynomial in size $\rightarrow$ longest paths then correspond to dual bounds.

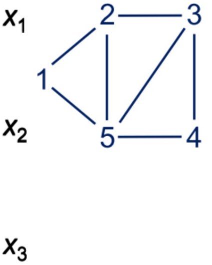
Have to merge also nodes for which states are not the same.

Research goal: Improve BDD construction mechanisms so that tighter bounds can be achieved with the same BDD size.

Top-Down Construction of Relaxed DD







To build **relaxed** DD, merge some additional nodes as we go along.

Take the **union** of merged states

To build **relaxed** DD, merge some additional nodes as we go along.

Take the **union** of merged states.

Considered BDD Construction

We consider

- top-down
- layerwise
- zero-suppressing long-arcs

construction of relaxed BDDs (Bergman et al. 2013) with max width β for the

- Maximum (Weighted) Independent Set Problem (MISP)
- Set Cover Problem (SCP)

Merging Strategies

Merging: Superimpose nodes in BDD while not losing feasible solutions.

Different merging strategies to keep layer within maximum width:

- Bulk Merging
- Iterative Merging

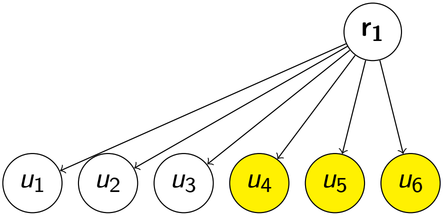
Minimum Longest Path (minLP) Value Bulk Merging

Bergman et al. (2013)

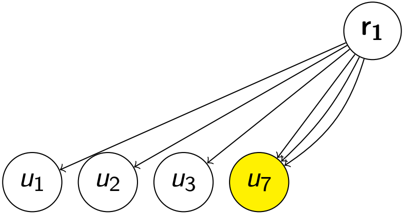
- Sort nodes in a layer descending by longest path length to them $z^{\mathrm{lp}}(u)$.
- Merge the tail into one node so that the maximum width is not exceeded.

Rationale: Merge nodes that are unlikely to be part of longest path.

Classical minLP Selection



Classical minLP Merging



minLP Ties

Example with $\beta = 10$.

<table>
<tr><td>rank(u)</td><td>1</td><td>1</td><td>3</td><td>4</td><td>4</td><td>4</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td><td>12</td><td>13</td><td>14</td><td>15</td></tr>
<tr><td>$z^{\mathrm{lp}}(u)$</td><td>9</td><td>9</td><td>8</td><td>7</td><td>7</td><td>7</td><td>6</td><td>6</td><td>6</td><td>6</td><td>6</td><td>5</td><td>4</td><td>3</td><td>1</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="5">$\underbrace{\qquad\qquad}_{T}$</td><td colspan="4">$\underbrace{\qquad\qquad}_{B}$</td></tr>
</table>

State Similarity

- Recall: State of nodes determines feasible completions.
- Define an informal **merging distance function** for pairwise merging of nodes u and v.
- Smaller distance between nodes should less likely increase the lengths of paths going through $u \oplus v$.

Initial motivation: Hamming distance between states.

Upper Bound from State

Idea: use upper bound directly from state $s(u)$ to estimate remaining longest path length starting from given node u.

For example, coarse upper bound in the MISP: cardinality of the set representing the state.

Merging Distance Functions

Given two nodes u, v in a layer with corresponding states $s(u), s(v)$.

1. Hamming: $d_H(u, v) = |s(u) \triangle s(v)|$.
2. Increase in upper bound:
   $d_{\mathrm{ub}}(u, v) = \max\{z^{\mathrm{ub}}(w) - z^{\mathrm{ub}}(u), z^{\mathrm{ub}}(w) - z^{\mathrm{ub}}(v)\}$.
3. Upper bound: $\tilde{d}_{\mathrm{ub}}(u, v) = z^{\mathrm{ub}}(w)$.

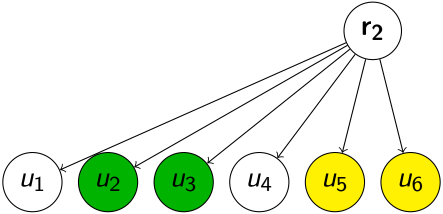
minLP/State Similarity Hybrid Merging Heuristic

minLP gives strong results. *Issue*: does not take similarity of states into account → introduce iterative pairwise state similarity merging gently.

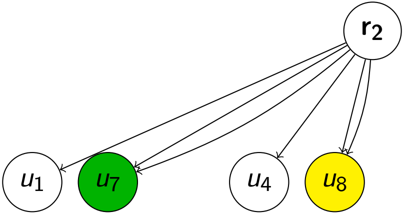
rank(u)	1	1	3	4	4	4	7	7	7	7	7	12	13	14	15
$\tilde{z}^{\mathrm{lp}}(u)$	1.	1.	.875	**.75**	**.75**	**.75**	**.625**	**.625**	**.625**	**.625**	**.625**	**.5**	0.375	.25	0.
									T					B	

Region T extensible to the left by parameter δ_l and to the right by parameter δ_r. If $\delta_l = \delta_r = 0$ we only consider ties.

minLP/State Similarity Hybrid Selection



minLP/State Similarity Hybrid Merging

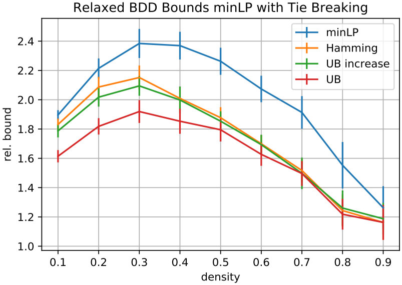


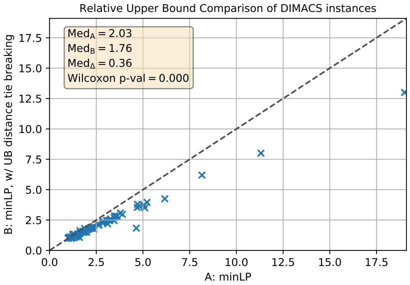
Combine Bulk and Iterative Merging

1. Bulk merge $\oplus B = w$.
2. Pairwise iterative merging over nodes $T \cup \{w\}$ choosing always the pair with minimum distance $d(u, v)$.

Pure Tie Breaking MISP

Strong results for small-width BDDs with pure tie breaking $\delta_l = \delta_r = 0$ for unweighted MISP on 180 random graphs by Bergman and DIMACS instance set:





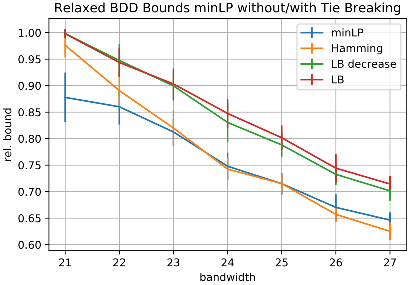
DIMACS Results

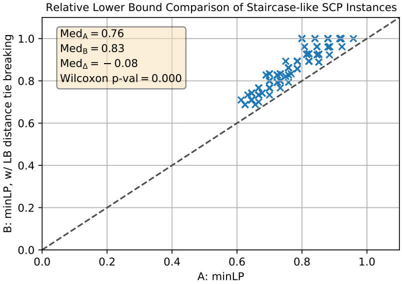
Table: Relative upper bounds of relaxed BDDs obtained with different merging heuristics and widths $\beta \in \{10, 100\}$ for selected DIMACS instances.

	$\beta = 10$				$\beta = 100$			
inst	minLP	d_H	d_{ub}	$\tilde{d}_{\mathrm{ub}}$	minLP	d_H	d_{ub}	$\tilde{d}_{\mathrm{ub}}$
brock200_1	2.29	2.14	2.14	**1.90**	1.81	**1.62**	1.67	1.67
C500.9	3.05	3.00	2.81	**2.47**	2.61	2.46	2.40	**2.28**
gen400_p0.9_55	2.25	2.13	2.04	**1.82**	1.91	1.82	1.80	**1.73**
keller4	1.91	**1.55**	1.64	**1.55**	1.45	**1.18**	**1.18**	**1.18**
MANN_a45	1.34	1.34	**1.21**	1.30	**1.08**	1.32	1.27	1.19
p_hat300-3	2.19	2.11	2.08	**1.86**	1.86	1.75	1.81	**1.69**
p_hat700-2	2.59	2.45	2.32	**2.18**	2.14	1.98	1.95	**1.93**

Pure Tie Breaking SCP

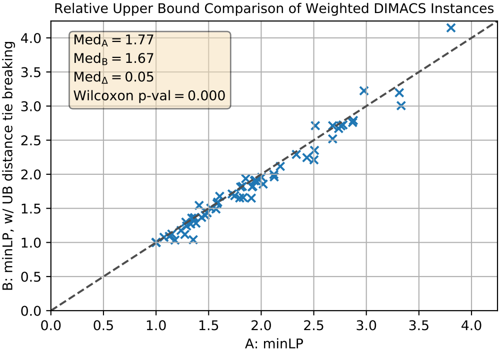
Median increase in the lower bound value of 0.08.





Weighted Problems

For weighted problems, ties are less likely to occur and $\delta_l = \delta_r = 0$ degenerates to minLP. Raced parameters $(0.185, 0.043)$ using irace on weighted DIMACS dataset, significant improvement but not so strong as before.



Conclusion

- Results for MISP and SCP indiciate that state similarity based merging works well together with minLP when ties occur naturally.
- Work needs to be done for weighted problems, where we have virtually no ties, to achieve a larger effect.

References

David Bergman, Andre A Cire, Willem-Jan van Hoeve, and John N Hooker.
Variable ordering for the application of BDDs to the maximum independent set problem.
In *International Conference on Integration of AI and OR Techniques in Constraint Programming*, volume 7298 of *LNCS*, pages 34–49. Springer, 2012.

David Bergman, Andre A Cire, Willem-Jan van Hoeve, and John N Hooker.
Optimization bounds from binary decision diagrams.
INFORMS Journal on Computing, 26(2):253–268, 2013.

ευχαριστώ πολύ!