Merging Quality Estimation for Binary Decision Diagrams with Binary Classifiers

LOD 2019, September 10-13, Certosa di Pontignano, Siena, Tuscany, Italy

Nikolaus Frohner and Günther R. Raidl

September 11, 2019

ALGORITHMS AND COMPLEXITY GROUP

Binary Decision Diagrams (BDDs)

Introduced by Lee in 1959 as compact representation of boolean functions and further elaborated on by Akers in eponymous "Binary decision diagrams" (1978) as rooted, directed, acyclic, multigraphs.

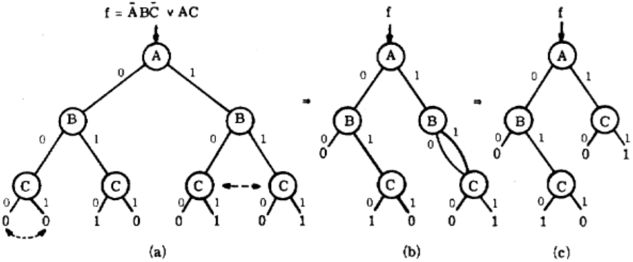


Fig. 4. Simplifying a diagram.

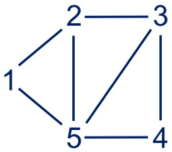
Figure: adapted from "Binary decision diagrams" by Akers, page 2

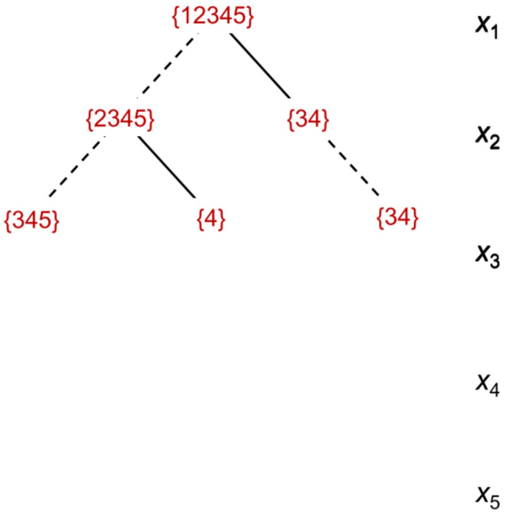
BDDs in Combinatorial Optimization

Introduced into the field of combinatorial optimization by Hadzic and Hooker (2006), for post-optimality analysis.

Representation of solution space where paths represent solutions with associated objective value and longest paths correspond to maxima.

Top-Down Construction of BDDs



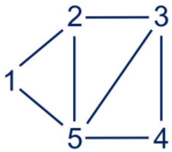


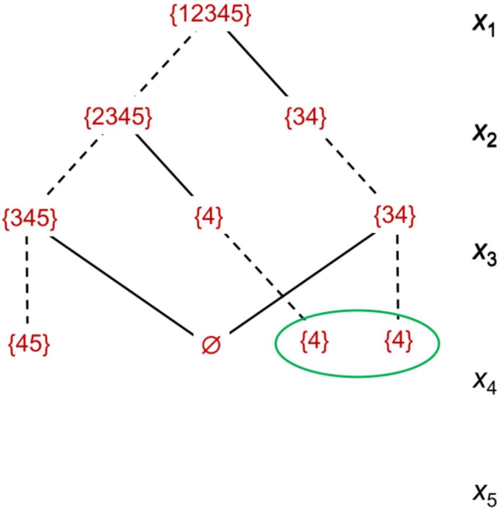
Exact DD for stable set problem

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

(from Hooker (2016))

Top-Down Construction of BDDs



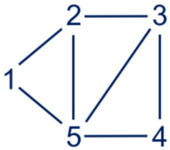


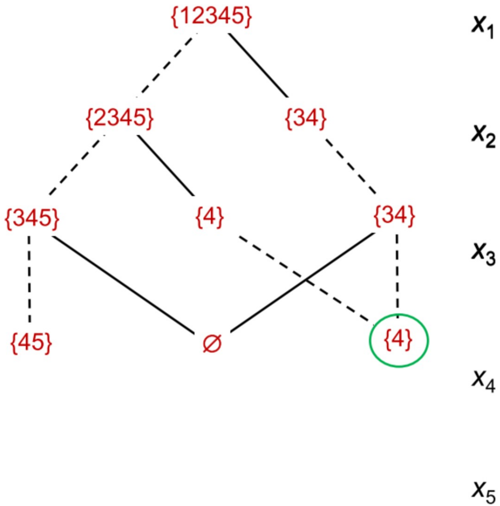
Exact DD for stable set problem

Merge nodes that correspond to the same state

(from Hooker (2016))

Top-Down Construction of BDDs





Exact DD for stable set problem

Merge nodes that correspond to the same state

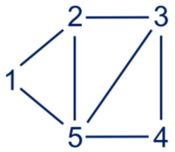
(from Hooker (2016))

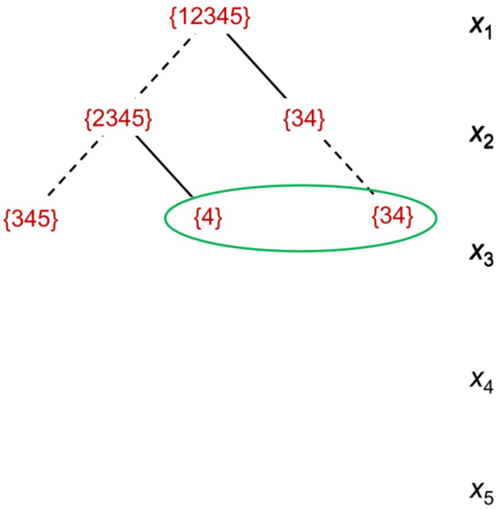
Relaxed BDDs

- Provides new means for relaxation, besides for example Linear Programming based or Lagrangian.
- Relaxed BDD represents superset of all feasible solutions.
- BDD kept compact by merging also nodes for which states are not the same $\rightarrow$ longest paths then usually correspond to upper bound for represented problem instance.

Decision of which nodes to merge is job of **merging heuristic**.

Top-Down Construction of Relaxed BDDs



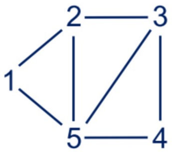


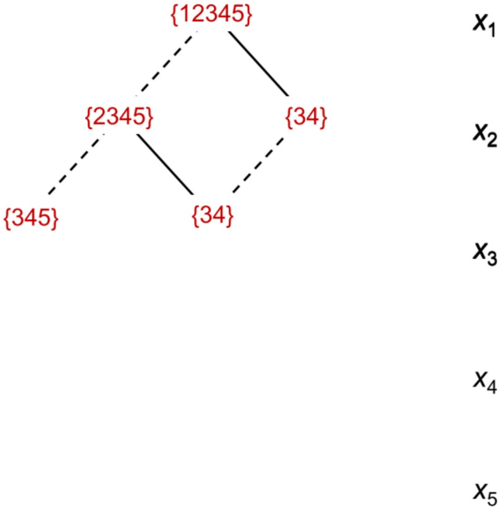
To build **relaxed** DD, merge some additional nodes as we go along.

Take the **union** of merged states

(from Hooker (2016))

Top-Down Construction of Relaxed BDDs



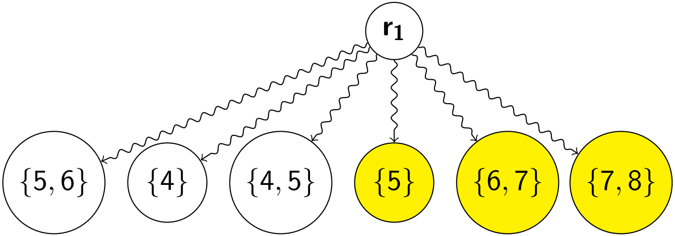


To build **relaxed** DD, merge some additional nodes as we go along.

Take the **union** of merged states.

(from Hooker (2016))

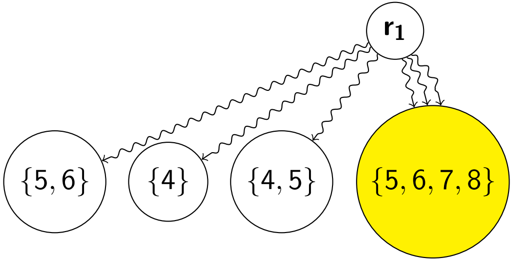
Classical minimum longest path (minLP) Selection



Order nodes in given layer by longest path (LP) length from root ($\mathbf{r_1}$).

States of nodes are represented by sets of elements that can still be selected.

Classical minLP Merging



Merges nodes from the back into one node.

May result into nodes with **large** states, yielding higher upper bounds for the resulting nodes, since more **infeasible paths** are likely to be introduced.

Similarity Based Merging

- We introduced minLP and state similarity based hybrid merging heuristic that improved bounds for small width BDDs for the Maximum Independent Set Problem (MISP) and the Set Cover Problem (SCP) (see our LION 13 paper) via tie breaking.
- Issue for weighted MISP instances, where ties are less likely to occur.

Merging Quality

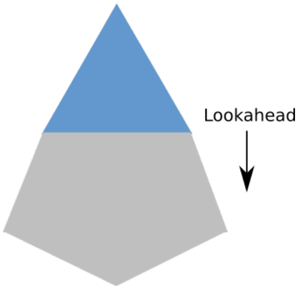
Idea: do not always (i.e., in each layer) apply the same merging heuristic, instead go for the "locally best" one out of a set of merging heuristics.

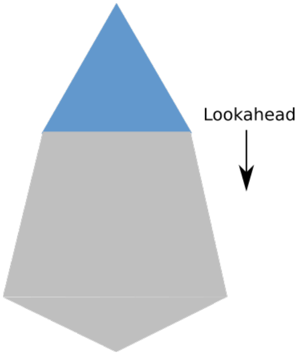
We define the locally best as: For which the completition of the decision diagram using minLP merging would result in the tightest bound.

Perfect Lookahead

Given the nodes of a layer that needs to be reduced in width:

- Apply all available merging heuristics, including minLP, on shallow BDD copy and finish construction using minLP.
- Finally, apply the merging heuristic that yielded best bound.





k-layers Lookahead

Lookahead used by Bergman et al. (2012) for dynamic variable ordering, we use it in the context of merging.

Perfect lookahead too expensive but gives us ground truth:

$f(H, H') = 1$, if H yields a strictly tighter bound than H', otherwise 0.

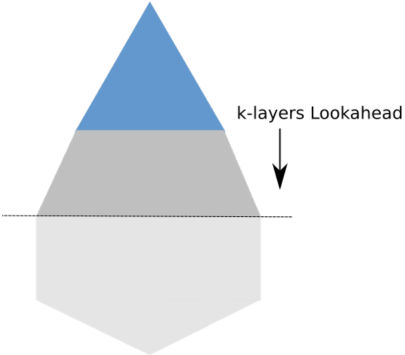
k-layers Lookahead

Only look k-layers ahead and gather p features by layer $\mathbf{Y}$ for two merging heuristics and estimate which one will result in a better final bound.

Binary classification function:

$\tilde{h}\colon \mathbb{R}^{p\times k} \times \mathbb{R}^{p\times k} \to [0,1]$.

$$h_\alpha(\mathbf{Y},\mathbf{Y}') = \begin{cases} 0, & \tilde{h}(\mathbf{Y},\mathbf{Y}') < \alpha \\ 1, & \tilde{h}(\mathbf{Y},\mathbf{Y}') \geq \alpha \end{cases}$$

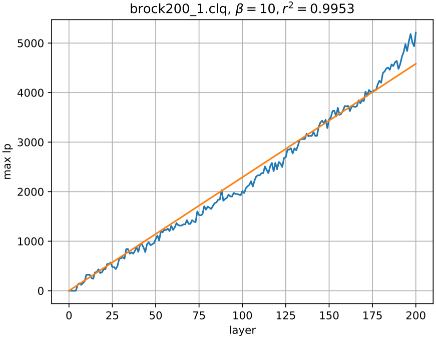


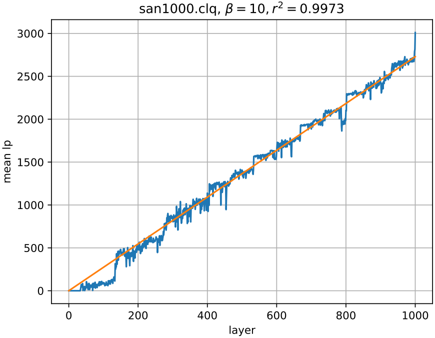
Possible Features

- min/mean/max of longest path values of nodes in layer
- min/mean/max of problem specific upper bound values of nodes in layer
- layer progress $l/l_{\max}$

Linear Approximation of Bound Growth

Noisy, linear growth of bounds (maximum longest path values) over layers.



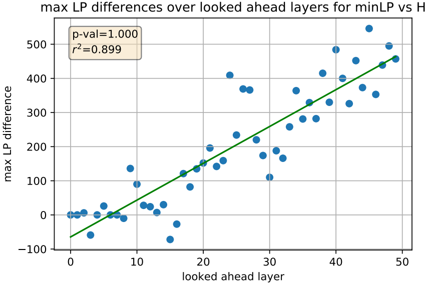


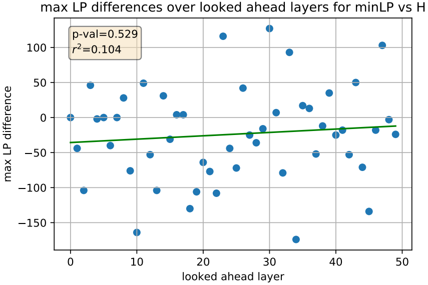
Classifiers

- Linear regression considering differences $\Delta\mathbf{Y} = \mathbf{Y}_{\mathrm{minLP}} - \mathbf{Y}_H$.
- max-maxLP: compare the maximum of the maximum of the longest path values over all looked-ahead layers.
- Wilcoxon signed rank sum test on paired features $\mathbf{Y}_{\mathrm{minLP}}, \mathbf{Y}_H$.
- Neural network based classifier.

LR Examples

- Left a true positive (minLP worse than H).
- Right a true negative (minLP not worse than H).





Train & Test Data

- Take random layer in random graph and apply hybrid merging with random parameters.
- Finish construction of BDD to see which one performs better, yielding the features and ground truth.
- Created 21000 training & test samples, approximately balanced, from 1000 random weighted graphs.

Neural Network Classifier

Use more features

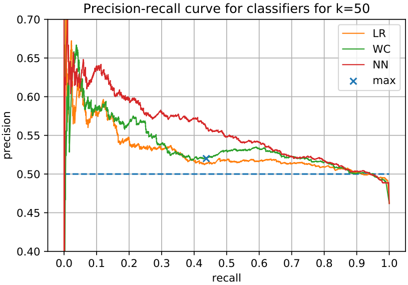
- max of the longest path values $z^{\mathrm{lp}}(u)$
- max of the upper bound values $z^{\mathrm{ub}}_{\mathrm{MISP}}(u)$
- layer progress $l/l_{\max}$

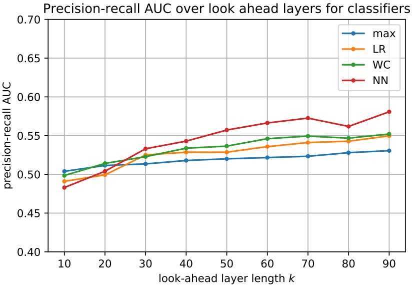
resulting in input dimension of $2k+1$

NN outputs value between 0.0 and 1.0.

Classifier Comparison

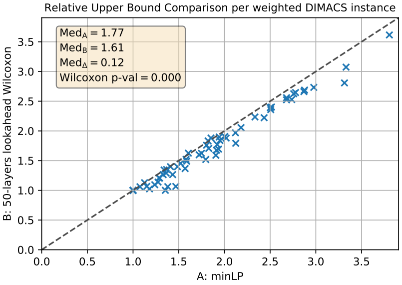
Precision-recall comparison on training data.

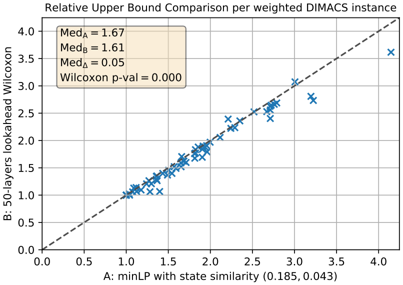




Classifiers on Weighted DIMACS

Baseline merging heuristic is minLP with tie breaking and the competing algorithm uses the raced parameter set $(0.185, 0.043)$, evaluated by $k = 50$ layers lookahead with Wilcoxon classifier.





Results on Weighted DIMACS[1]

Weighted MISP relative bound improvements for different classifiers, parameters, and compared to different baseline approaches.
$\widetilde{\Delta}, \overline{\Delta}$: median/mean improvement of relative bounds.

comparing approach	PLA		k	max		LR		WC		NN	
	$\widetilde{\Delta}$	$\overline{\Delta}$		$\widetilde{\Delta}$	$\overline{\Delta}$	$\widetilde{\Delta}$	$\overline{\Delta}$	$\widetilde{\Delta}$	$\overline{\Delta}$	$\widetilde{\Delta}$	$\overline{\Delta}$
pure minLP	0.16	0.17	30	0.09	0.11	0.07	0.08	0.09	0.11	0.11	0.11
minLP with state similarity	0.09	0.11		0.04	0.06	0.02	0.03	0.04	0.06	0.04	0.06
pure minLP	0.16	0.17	50	0.09	0.11	0.09	0.11	0.12	0.13	0.12	0.13
minLP with state similarity	0.09	0.11		0.03	0.06	0.03	0.06	0.05	0.08	0.08	0.08
pure minLP	0.16	0.17	70	0.10	0.12	0.10	0.12	0.12	0.14	0.15	0.16
minLP with state similarity	0.09	0.11		0.04	0.06	0.03	0.07	0.05	0.09	0.08	0.11

[1] https://github.com/jamestrimble/max-weight-clique-instances/tree/master/DIMACS

Conclusion & Future Work

We could improve relaxed BDD bounds of weighted MISP instances using lookahead mechanism as compared to pure classic minLP merging heuristic.

- Main issue: computationally very expensive.
- Search for stronger features to identity "locally best" merging heuristic with less effort.
- Improve classification to reduce lookahead length.
- Test with reduced BDD width for lookahead.
- Test on other problems, weighted set cover problem as next goal.

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 Artificial Intelligence (AI) and Operations Research (OR) Techniques in Constraint Programming*, 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.

Nikolaus Frohner and Günther R. Raidl.
Towards improving merging heuristics for binary decision diagrams.
In Nikolaos F. Matsatsinis, Yannis Marinakis, and Panos Pardalos, editors, *Learning and Intelligent Optimization – 13th International Conference, LION 13*, volume 11968 of *LNCS*, pages 30–45. Springer, 2019.
URL `https://www.ac.tuwien.ac.at/files/pub/frohner-19a.pdf`.

Thank you