On the Impact of Subproblem Orderings on Anytime AND/OR Best First Search

William Lam, Kalev Kask, Rina Dechter[1], Javier Larrosa[2]

[1]University of California, Irvine, USA, [2]UPC Barcelona Tech, Spain

INTRODUCTION

Best-first search can be regarded as an anytime scheme for producing lower bounds on the optimal solution, a characteristic that is mostly overlooked. We explore this topic in the context of AND/OR best-first search (AOBF), guided by the mini-bucket elimination (MBE) heuristic, when solving graphical models. In that context, the impact of the secondary heuristic for subproblem ordering may be significant, especially in the anytime context.

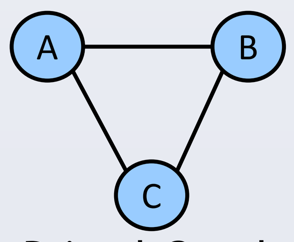
We illustrate in this paper that the new concept of *bucket errors* can advise in providing effective subproblem orderings in AND/OR search.

BACKGROUND

Graphical Model

- $\boldsymbol{M} = (\boldsymbol{X}, \boldsymbol{D}, \boldsymbol{F})$
- $\boldsymbol{X} = \{X_i : i \in V\}$, a set of variables indexed by a set V
- $\boldsymbol{D} = \{D_i : i \in D\}$ is the set of finite domains of values for each X_i
- $\boldsymbol{F} = \{f_\alpha :\ \alpha \in F\}$ is a set of discrete functions, where $\alpha \subseteq V$ and $X_\alpha \subseteq X$ is the scope of f_α

$$F(A,B,C) = f(A,B) + f(A,C) + f(B,C)$$



Primal Graph

Our focus: computing

$$min_{\mathbf{X}} \sum_{\alpha \in F} f_\alpha(X_\alpha)$$

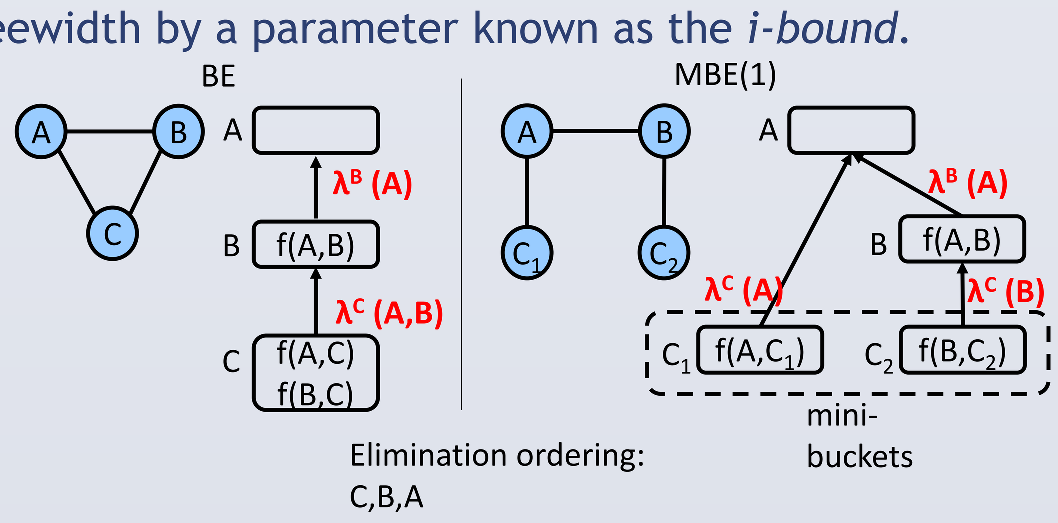
and its assignment

Bucket Elimination (BE) [Dechter 1999]

- Solves the min-sum problem by eliminating variables one at a time.
- Complexity: exponential in the *induced (tree) width* of the underlying primal graph

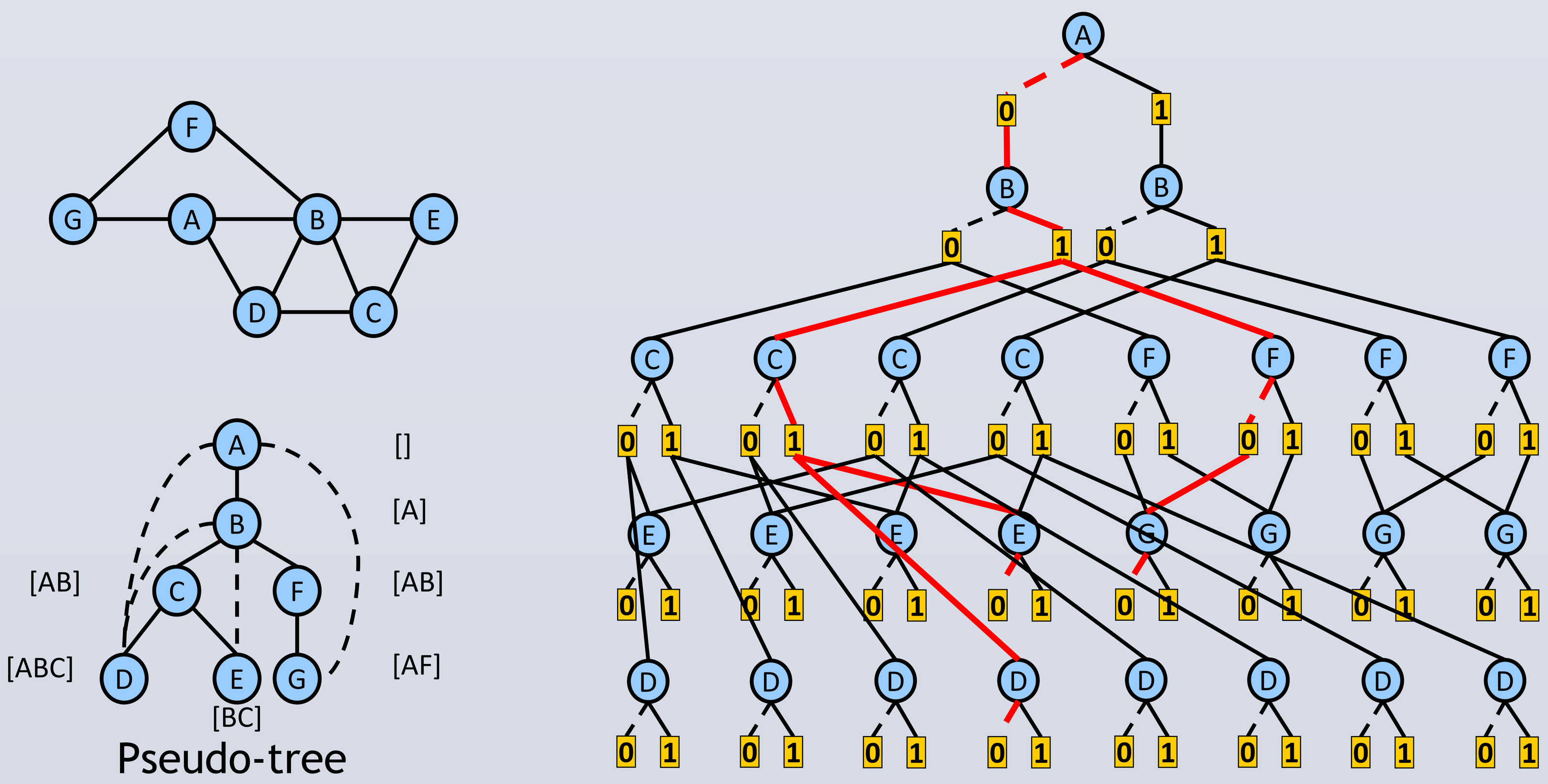
Mini-Bucket Elimination (MBE) [Dechter and Rish 2003]

- We can *approximate* BE by solving a relaxation created by duplicating variables to bound the treewidth by a parameter known as the *i-bound*.



AND/OR Search

- The AND/OR search space of a graphical model is guided by the decomposition suggested by the pseudo-tree.



Generalized Best-First Search for AND/OR Graphs [Pearl 1984]

1. Identify the most promising partial solution tree *t* with an evaluation function f_1
2. Expand nodes with evaluation function f_2

<u>Most work deals with choosing f_1, while f_2 is chosen ad-hoc.</u>

AND/OR Best-First Search [Marinescu 2009]

- A state-of-the-art algorithm that is a variant of the AO* algorithm, specialized for AND/OR search spaces over graphical models
- Maintains the best partial solution tree *t* via steps to *expand* and *revise* it
- Uses f_1 (MBE heuristic) as the ad-hoc choice for f_2
- We explore this in the context of generating <u>*anytime lower-bounds*</u> in this work

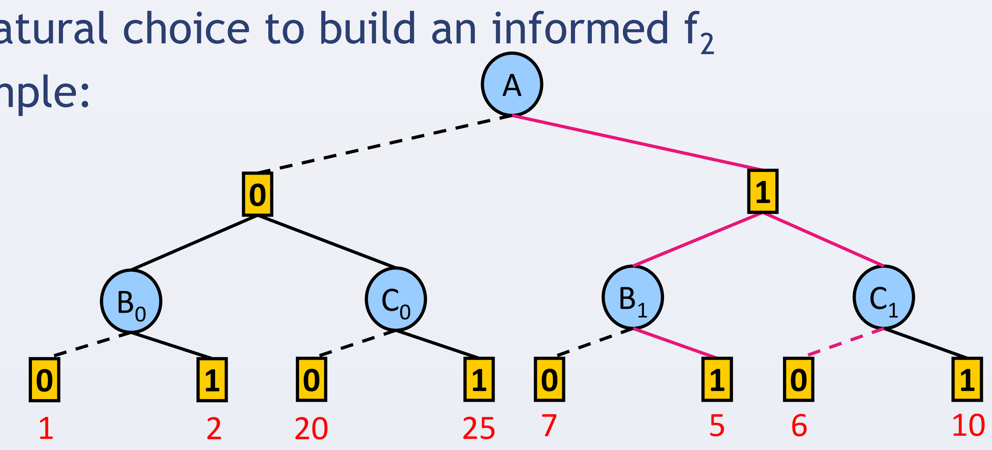
Bucket Error [Dechter et al. 2015]

- The *difference* between the function-message that would have been computed in an *individual* bucket without partitioning and the message computed by MBE.
- Example using BE and MBE figure above: **$Err_C(A,B) = \lambda^C(A,B) - (\lambda^C(A) - \lambda^C(B))$**
- Captures the *local* error induced by the MBE partitioning
- Equivalent to *depth-1 lookahead residual*

IMPACT OF SUBPROBLEM ORDERING

- Given $f_1(t) < C^*$, the selection of which subproblem to expand next (f_2) can influence the number of nodes expanded
- Intuitively, we want to expand the subproblem that leads to the *greatest increase* in f_1 therefore discovering as soon as possible that the optimal extension to *t* is >C*
- The *increase* is equivalent to the *look-ahead residual*, making *bucket errors* a natural choice to build an informed f_2

Example:



Optimal cost C* = 11,
Solution tree highlighted in red

- Left-first subproblem ordering: the entire search space is expanded
- Right-first subproblem ordering: the space rooted by B_0 is not expanded

BUCKET ERRORS FOR AND/OR BEST-FIRST SEARCH

- Bucket errors are *1-level residuals*, so this only tells us about the increase f_1 for the immediate next expansion
- We introduce the notion of *subtree error* to capture *d-level residuals* to tell us exactly about increase in f_1 beyond the next node
- Equivalent to solving the problem in complexity to compute: so we approximate by summing bucket errors

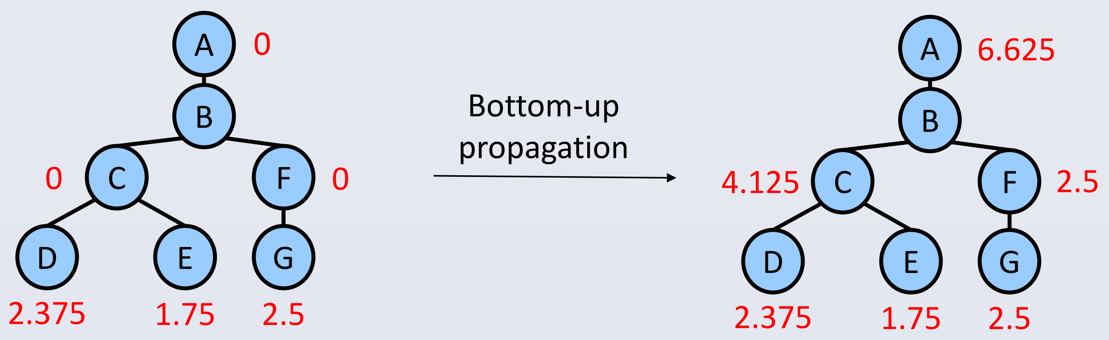


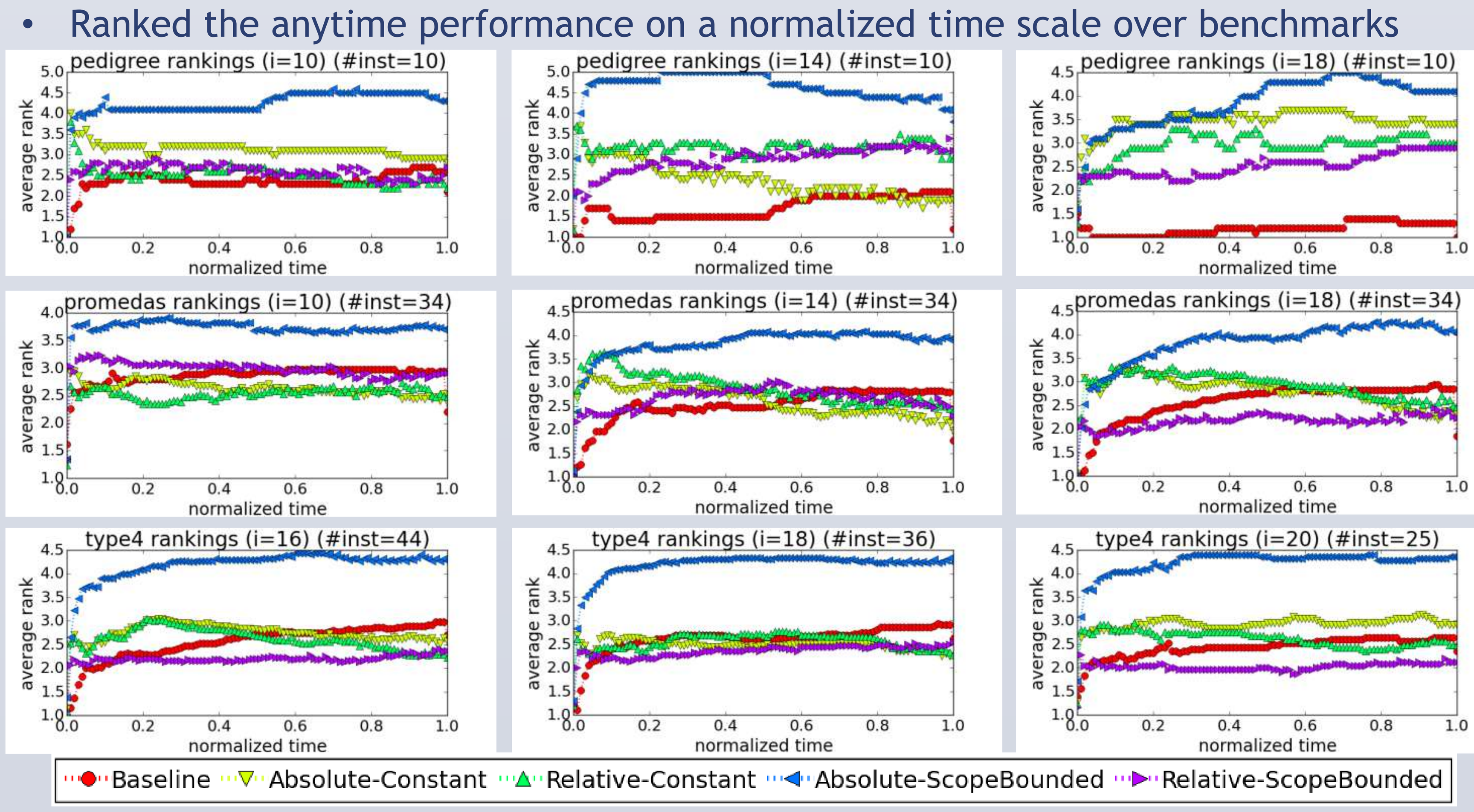
Illustration of subtree error calculation when bucket errors are *constants*

- We consider two types of representations of bucket errors for each variable – these are done for computational tractability
- ***Constant***: computed by averaging over all instantiations of the error function
- ***Scope-bounded***: computed by averaging over all instantiations over a *subset* of variables in the error function
- We also consider two types of error values in the original functions
- ***Absolute*** bucket errors: standard definition of bucket error
- ***Relative*** bucket errors: divide the absolute error by the exact message value
- Total of <u>***4***</u> variants of our f_2 (***constant/scope-bounded***, ***absolute/relative***)

EXPERIMENTS AND CONCLUSION

We ran AND/OR Best-First (AOBF) search using the MBE-MM heuristic [Ihler et al. 2012]

- Tested various *i-bounds* to vary heuristic strength
- Compare 4 variants against the ad-hoc baseline
- Ranked the anytime performance on a normalized time scale over benchmarks



- The baseline is better on the *pedigree* instances, while the *promedas* and *type4* instances, our methods can outrank the baseline
- This is directly related to the heuristic strength on the problem – low errors are uninformative for *pedigrees*; the higher errors in *promedas/type4* are informative
- **Our results show that subproblem ordering is important when errors are high and should be applied to any type of AND/OR Best-First search.**

REFERENCES

Rina Dechter. 'Bucket elimination: A unifying framework for reasoning', Artificial Intelligence 113:41–85, (1999)

Rina Dechter and Irina Rish. 'Mini-buckets: A general scheme for bounded inference', JACM 50(2): 107-153, (2003).

Judea Pearl. 'Heuristics: Intelligent Search Strategies', Addison-Wesley, 1984

Radu Marinescu and Rina Dechter, 'Memory intensive and/or search for combinatorial optimization in graphical models', Artif. Intell., 173(16- 17), 1492–1524, (2009).

Rina Dechter, Kalev Kask, William Lam, and Javier Larrosa, 'Lookahead with mini-bucket heuristics for mpe', in AAAI, (2016).

Alexander Ihler, Natalia Flerova, Rina Dechter, and Lars Otten, 'Join-graph based cost-shifting schemes', in UAI, (2012).

Supported in part by NSF grants IIS-1526842 and IIS-1254071, the US Air Force under Contract No. FA8750-14-C- 0011 under the DARPA PPAML program, and MINECO/FEDER under project TIN2015-69175-C4-3-R.