Empowering Mini-Bucket in Anytime Heuristic Search with Look-Ahead: Preliminary Evaluation

William Lam, Kaleb Kask, and Rina Dechter
University of California, Irvine
{willmlam,kkask,dechter}@ics.uci.edu

Abstract
The paper explores the potential of look-ahead methods within the context of AND/OR search in graphical models using the Mini-Bucket heuristic for combinatorial optimization tasks (e.g., weighted CSPS or MAP inference). We study how these methods can be used to compensate for the approximation error of the initially generated Mini-Bucket heuristics, within the context of anytime Branch-And-Bound search.

Introduction
This paper investigates the effect of look-ahead on the anytime performance of search solving the combinatorial optimization task of MPE in graphical models when using the mini-bucket heuristic (Dechter, Kask, and Lam 2015). There, we introduced the approach and showed a relationship between the residual of the mini-bucket heuristic and a newly defined notion of bucket-error. Look-ahead can be carried out to varying levels during a search process. When we have well behaved monotone heuristics, a more aggressive look-ahead yields a more accurate heuristic but requires more time. This paper builds on our recent work on look-ahead in which we developed the notion of a bucket-error. Look-ahead in which we developed the notion of a bucket-error.

Contributions. Look-ahead can be guided by the decomposition that is suggested by the pseudo-tree, yielding an AND/OR search space. In the rest of the paper we will use the following notation.
- $B_Y$ is the bucket of variable $Y$.
- $\psi_Y$ - sum of all the original functions in $B_Y$. $\psi_Y^p$ - sum of those in the $p$th mini-bucket $MB_Y^p$ of $B_Y$.
- $\bar{X}_{1..p}$ - A path of variables from $X_1$ to $X_p$. $\bar{x}_{1..p}$ - An assignment of values to the variables in $X_{1..p}$.

MPE Look-ahead for Search
Since the AND/OR search space is guided by the pseudo-tree many of the concepts of look-ahead can be viewed relative to the pseudo-tree as well. A d-level look-ahead computation can be organized over variables of the pseudo-tree $T$ that are within distance $d$ of the current variable $X_p$.

**Definition 1** (d-level subtree $T_{(X_p,d)}$) Given a pseudo-tree $T$ and an integer $d$, $T_{(X_p,d)}$ is the set of variables in the $d$-level subtree of $T$ rooted at $X_p$.

**Definition 2** (Look-ahead MBE-heuristic in AND/OR) Given an AND/OR search space guided by a pseudo-tree $T$ and a corresponding MBE heuristic function $h^{mbe}$, the...
**d-level look-ahead heuristic of** $h_{mbe}$ **at** $X_p$ **is a function, whose scope is subsumed in** $\bar{X}_{1..p}$, **denoted** $h_{X_p}^{lh}(s_p)$, $s_p = \bar{x}_{1..p}$:

- $h_{X_p}^{lh}(0)(s_p) = h_{X_p}^{mbe}(s_p)$
- $h_{X_p}^{lh}(d)(s_p) = \sum_{Y \in \text{child}(X_p)} \min(\psi_Y(s_p, y) + h_{Y}^{lh}(d-1)(s_p, y))$

**The d-level residual at** $X_p$ **is**

- $res_{X_p}^{lh}(d)(s_p) = h_{X_p}^{lh}(d)(s_p) - h_{X_p}^{mbe}(s_p)$

**Experiments**

We ran AND/OR Branch and Bound (AOBB) (Marinescu and Dechter 2009) using the state-of-the-art MBE-MM heuristic (Ihler et al. 2012) on 3 different benchmarks with a time bound of 6 hours for each instance. To evaluate the general impact of look-ahead across a range of depths $d$, we collected the solutions reported by our solver at various time snapshots and reported those in instance by instance anytime graphs.

**Figure 1** plots the time vs. solution quality for an instance from each benchmark. Points that are higher and to the left indicate superior anytime performance. If a line runs off the right end of the plot, it indicates that the problem was not solved within the time bound. Each row of the figure corresponds to a particular problem instance and the columns correspond to different i-bounds. In this small sample of our results, we see indeed that the look-ahead scheme has a positive impact on anytime performance at earlier times, especially when the i-bound is lower. However, as the search time and/or i-bound increase, shallower look-ahead depths can become more preferable. In results on exact search (Dechter, Kask, and Lam 2015), a look-ahead depth of 2-3 often performed the best even for the same set of lower i-bounds, with deeper look-ahead yielding decreased performance due to time overhead. Here, we see that deeper look-ahead can give superior early anytime performance.

**Conclusion**

We performed a preliminary evaluation of the effect of look-ahead on the anytime performance of solving MPE. Our results show that look-ahead can be a cost effective method of achieving higher quality solutions sooner, especially when the heuristic is weak. The search time is also a factor, suggesting that it may also be beneficial to dynamically change the look-ahead depth over time. Future work in estimating the residual and its relative effect during search can guide us towards more flexible look-ahead subtrees for this purpose.

**References**


