# Anytime AND/OR Depth-first Search for Combinatorial Optimization (Extended Abstract)*

Lars Otten[1] and Rina Dechter[2]

[1] University of California, Irvine** – lotten@uci.edu
[2] University of California, Irvine – dechter@ics.uci.edu

**Abstract.** One popular and efficient scheme for solving combinatorial optimization problems over graphical models *exactly* is depth-first Branch and Bound. However, when the algorithm exploits problem decomposition using AND/OR search spaces, its anytime behavior breaks down. This article 1) analyzes and demonstrates this inherent conflict between effective exploitation of problem decomposition (through AND/OR search spaces) and the anytime behavior of depth-first search (DFS), 2) presents a new search scheme to address this issue while maintaining desirable DFS memory properties, and 3) analyzes and demonstrates its effectiveness through comprehensive empirical evaluation. Our work is applicable to *any* problem that can be cast as search over an AND/OR search space.

## 1 Introduction

Max-product problems over graphical models, generally known as MPE (most probable explanation) or MAP (maximum a posteriori) inference, have many applications with practical significance, ranging from computational biology and genetics to scheduling tasks and coding networks. One established and efficient class of algorithms for solving these problems exactly is depth-first Branch and Bound over AND/OR search spaces. Developed in the past decade within the probabilistic reasoning and constraint communities, these methods are effective because they use sophisticated lower bound schemes such as soft arc-consistency [2] or the mini-bucket heuristic [3, 4], because they avoid redundant computation using caching schemes, and most significantly, because they take advantage of problem decomposition by exploring an AND/OR search space [5] or an equivalent representation. The efficiency of these algorithms was established in several evaluations, including recent UAI competitions [6], and their properties when used for exact computation are well documented [7, 4, 8].

A principled alternative is presented by best-first schemes, but while provably superior in terms of number of node expansions, these often fail when a problem has large induced width due to the generally exponential size of the algorithm's OPEN list; moreover, they can only provide a solution at termination [8]. Depth-first search is therefore often preferred because of its flexibility in working with bounded memory – the OPEN list of nodes grows linearly – and because of its *anytime behavior*. Namely, when finding a feasible solution is easy but an optimal one is hard, depth-first Branch and Bound

** Now at Google Inc. (ottenl@google.com)
* This is a summary of the full article published in AI Communications, Volume 25(3) [1].

generates solutions that get better and better over time, until it eventually discovers an optimal one. Thus it can function also as an approximation scheme for otherwise infeasible problems or when time is limited [9].

Indeed, in the 2010 UAI Approximate Inference Challenge participating Branch and Bound solvers performed competitively with respect to approximation (placing 1st and 3rd in some categories). But we also observed an inability to produce even a single solution on some instances, especially when the time bound was small. Thus motivated, this article demonstrates that the issue is rooted in the underlying AND/OR search space.

These search spaces were originally introduced to graphical models to facilitate problem decomposition during search (e.g. [10]) and can be explored by any search strategy. When traversed depth-first, however, all but one decomposed subproblem will be *fully solved* before a single overall solution can be composed, voiding the algorithm's anytime characteristics.

This article's main contribution is a new Branch and Bound scheme over AND/OR search spaces, called *Breadth-Rotating AND/OR Branch and Bound (BRAOBB)* that addresses the anytime issue in a principled way, while maintaining the favorable complexity guarantees of depth-first search. The algorithm combines depth-first and breadth-first exploration by periodically rotating over the different subproblems, each of which is processed depth-first.

Experimental evaluation is conducted on a variety of benchmark domains, including haplotype computation problems in genetic pedigrees, random grid networks, and protein side-chain prediction instances. We compare BRAOBB against one of the best variants of AND/OR branch and Bound search, AOBB [4], and against an "ad hoc" fix that we suggest – the latter algorithm relies on a heuristic to quickly find a solution to each subproblem before reverting to depth-first search. We furthermore compare against a state-of-the-art stochastic local search solver, which is specifically targeted at anytime performance but cannot provide any proof of optimality [11].

The empirical results demonstrate superior anytime behavior of BRAOBB, especially over problematic cases where standard AOBB and its ad hoc fix fail, including several very hard instances from the 2010 UAI Approximate Inference Challenge that were made available and three weighted constraint satisfaction problem instances that are known to be very complex. We also show how combining local search and exhaustive AND/OR search lets us enjoy the benefits of both approaches. Notably, a solver based on this concept recently won all three categories (20 seconds, 20 minutes, and 1 hour) in the MPE track of the PASCAL 2011 Inference Challenge [12, 13], the successor to the 2010 UAI Challenge.

## 2 Brief Overview of Results

As noted above, in AND/OR search spaces depth-first traversal of a set of independent subproblems will solve to completion all but one subproblem before the last one is even considered. As a consequence, the first generated overall non-optimal solution contains conditionally optimal solutions to all subproblems but the last one. Furthermore, depending on the problem structure and the complexity of the independent subproblems,

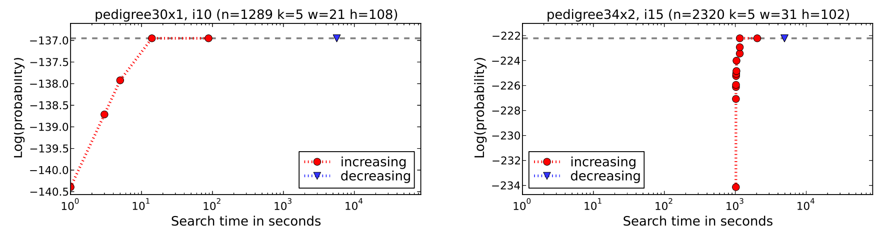


**Fig. 1.** Anytime performance of AOBB for differing subproblem oderings. Specified for each network: number of variables $n$, max. domain size $k$, induced width $w$ along the chosen ordering, height of the corresponding pseudo tree $h$. The dashed gray line indicates the optimal solution.

the time to return even this first non-optimal overall solution can be significant, practically negating the anytime behavior of depth-first search (DFS).

To illustrate, consider Figure 1, which depicts the anytime performance (best-known solution cost over time) of AOBB on two example problem instances. For demonstration purposes we apply a simple heuristic which has AOBB consider independent subproblems by increasing or decreasing hardness, based on the subproblem induced width. If decomposition yields only one large subproblem and several smaller ones, the latter can be solved depth-first in relatively little time, to be then combined with the incrementally improving solutions of the larger subproblem. This is exemplified by applying the "increasing" order to pedigree30x1, which has one hard subproblem and several other, simple ones: we see a suboptimal overall solution right away which is gradually improved upon; using the "decreasing" order AOBB spends a long time on solving the hardest subproblem to completion before returning any overall solution.

In case of pedigree34x2, however, decomposition yields two complex subproblems: the increasing subproblem order still outperforms its inverse, yet it returns the initial solution only after about 1,000 seconds. In fact, no possible subproblem ordering can lead to acceptable anytime behavior in this case due to the structure of subproblems, clearly highlighting the limits of this approach.

### 2.1 Breadth-Rotating AND/OR Branch and Bound (BRAOBB)

To remedy this issue, BRAOBB combines depth-first exploration with the notion of "rotating" through different subproblems in a breadth-first manner. Namely, node expansion still occurs depth-first as in standard AOBB, but the algorithm takes turns in processing subproblems, each up to a given number of operations at a time, round-robin style. This lets us develop all branches of the solution tree "simultaneously".

More systematically, the algorithm maintains a list of currently open subproblems and repeats the following high-level steps until completion:

1. Move to next open subproblem $P$ in a breadth-first fashion.
2. Process $P$ depth-first, until either:
   (a) $P$ is solved optimally,
   (b) $P$ decomposes into child subproblems, or

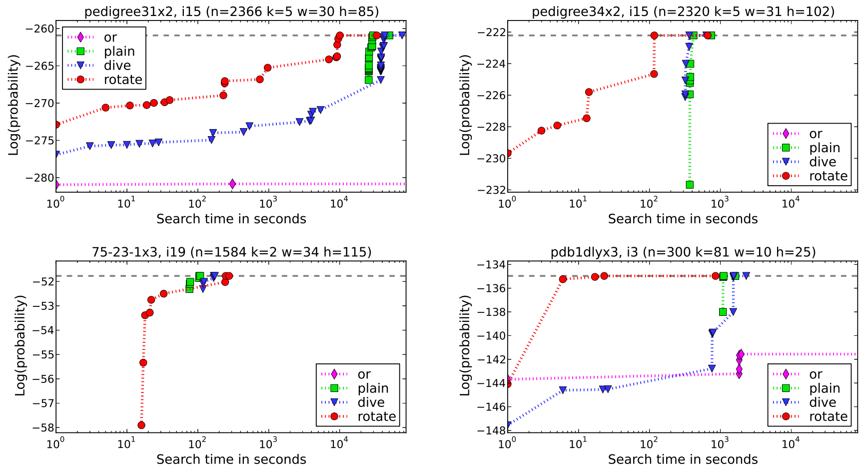


**Fig. 2.** Anytime performance of BRAOBB ("rotate") compared against "plain" AOBB and two other schemes (OR search and AOBB with "dive" extension, as outlined in the full article).

(c) a predefined threshold number of operations is reached.

The threshold in (c) is needed to ensure the algorithm does not get stuck in one large subproblem where the other two conditions, (a) and (b), do not occur for a long time. Furthermore, in order to focus on a single solution tree at a time, a subproblem is only considered "open" if it does not currently have any child subproblems. More details, algorithm pseudo code, and theoretical analysis are given in the full article [1].

Figure 2 shows four examples for the anytime perfomance of BRAOBB. For reference the plots also include AOBB and plain OR search, as well as AOBB with a "dive" extension (which performs an initial greedy dive into each subproblem – details in the full article). From the results it is clear that BRAOBB holds a decisive advantage over the other schemes evaluated here. It generally returns a first solution quickly and is consistently the first scheme, or one of the first, to do so. Furthermore, in almost all cases it proceeds to improve upon the initial solution quickly, again outperforming other schemes in the evaluation.

The full article also contains a number of summary statistics, for instance showing that after 5 seconds BRAOBB has found an initial solution for 510 out of 543 problem instances, compared to 269 for plain AOBB. And after 1 minute, BRAOBB has found the optimal solution for 321 instances compared to 274 for plain AOBB – again, refer to the full article for more details [1].

Moreover, the article also provides analysis of BRAOBB from several angles, including complexity analysis that shows that BRAOOB retains the favorable asymptotic guarantees of "plain" AND/OR search.

## References


1. Otten, L., Dechter, R.: Anytime AND/OR depth-first search for combinatorial optimization. AI Communications **25**(3) (2012)
2. Larrosa, J., Schiex, T.: Solving weighted CSP by maintaining arc consistency. Artif. Intell. **159**(1-2) (2004) 1–26
3. Dechter, R., Rish, I.: Mini-buckets: A general scheme for bounded inference. Journal of the ACM **50**(2) (2003) 107–153
4. Marinescu, R., Dechter, R.: AND/OR Branch-and-Bound search for combinatorial optimization in graphical models. Artif. Intell. **173**(16-17) (2009) 1457–1491
5. Nilsson, N.: Artificial Intelligence: A New Synthesis. Morgan Kaufmann, San Francisco, CA (1998)
6. Elidan, G., Globerson, A.: UAI 2010 approximate inference challenge. http://www.cs.huji.ac.il/project/UAI10/
7. Jégou, P., Terrioux, C.: Decomposition and good recording for solving max-CSPs. In: Proc. of the 16th Eureopean Conference on Artificial Intelligence (ECAI 2004), IOS Press, Amsterdam (2004) 196–200
8. Marinescu, R., Dechter, R.: Memory intensive AND/OR search for combinatorial optimization in graphical models. Artif. Intell. **173**(16-17) (2009) 1492–1524
9. Zilberstein, S.: Using anytime algorithms in intelligent systems. AI Magazine **17**(3) (1996) 73–83
10. Dechter, R., Mateescu, R.: AND/OR search spaces for graphical models. Artif. Intell. **171**(2-3) (2007) 73–106
11. Hutter, F., Hoos, H.H., Stützle, T.: Efficient stochastic local search for MPE solving. In: Proc. of the 19th International Joint Conference on Artificial Intelligence (IJCAI 2005), Professional Book Center, Denver, CO (2005) 169–174
12. Elidan, G., Globerson, A., Heinemann, U.: PASCAL 2011 probabilistic inference challenge. http://www.cs.huji.ac.il/project/PASCAL/
13. Otten, L., Ihler, A., Kask, K., Dechter, R.: Winning the PASCAL 2011 MAP Challenge with Enhanced AND/OR Branch-and-Bound