Proofs for the paper: Average Case Complexity of Branch-and-Bound Algorithms on Random b-ary Trees

Adrian Kügel

Faculty of Engineering and Computer Sciences, Ulm University, 89069 Ulm
Adrian.Kuegel@uni-ulm.de

1 Constant Probability of an Optimal Lower Bound

Lemma 1. *When using a best-first search Branch-and-Bound algorithm which breaks ties among nodes with equal lower bounds by selecting the node with greater depth, at most one node in each depth needs to be visited for which the calculated lower bound is optimal.*

Proof. Lets assume that there are two nodes u and v in depth d for which the calculated lower bound is optimal, and u is visited first among all nodes in depth d with optimal lower bound. This implies that the lower bound of v is not smaller than the lower bound of u. A best-first search Branch-and-Bound algorithm only visits nodes with lower bounds not greater than the global minimum. It follows that the lower bound of u must be equal to the global minimum, and at least one leaf node in the sub-tree of u has this minimum value. If the best-first search strategy breaks ties among nodes with equal lower bound by selecting a node with greater depth, all nodes in the subtree of u with lower bound values smaller or equal to the global minimum will be visited before node v, therefore also the leaf node with the global minimum value will be visited before node v. It follows that node v will not be visited, since a leaf node with the global minimum value has already been found. □

$$\begin{aligned} T(n) &= 1 \\ T(d) &= 1 + b \cdot (1-p) \cdot T(d+1) \end{aligned} \tag{1}$$

Explanation how this recurrence relation is derived: The recurrence relation only deals with nodes which do not lie on the path from the root node to the leaf node where the search stops. For each child node of a visited node, there are two cases: it can have an optimal lower bound, or it can have a sub-optimal lower bound. If it has a sub-optimal lower bound (which happens with probability $(1-p)$), the node has to be visited in the worst case. If it has an optimal lower bound (which happens with probability p), the node will not be visited, because an optimal lower bound implies that the lower bound value is greater or equal to the global minimum value. If the node would have a lower bound value equal to the global minimum value, it will either lie on the path from the root node to the leaf node with the minimum value (and thus, it should not be considered for the recurrence relation), or it will not be visited according to Lemma 1.

Theorem 1. *The expected number of visited nodes is in* $\mathcal{O}(n^2)$ *if* $b \cdot (1-p) \leq 1$*.*

Proof. According to Lemma 1, in each depth there will be at most one visited node for which an optimal lower bound was calculated. Therefore, an upper bound for the expected number of visited nodes is $\sum_{i=0}^{n} T(i)$. Let $b \cdot (1-p) \leq 1$.

$$\sum_{i=0}^{n} T(i) \leq \sum_{i=0}^{n} n - i + 1 \leq (n+1)^2 = \mathcal{O}(n^2)$$

□

2 Probability Distribution of Leaf Node Values and Lower Bound Values

Since a node is only visited if its lower bound is smaller than the minimum value of a leaf node visited so far, the condition for a node to be visited can be formulated as follows:

$$Y_{d,i} < \min_{0 \leq k < i \cdot b^{n-d}} \{X_{n,k}\} \Leftrightarrow Y_{d,i} < \min_{0 \leq k < i} \{X_{d,k}\} \tag{2}$$

In the next step we derive a formula for the expected number of nodes which are visited by a depth-first search Branch-and-Bound algorithm. We start by calculating the probability that the i-th node in depth d is visited.

$$\begin{aligned} & P\left(Y_{d,i} < \min_{0 \leq k < i} \{X_{d,k}\}\right) \\ = & \sum_{j=1}^{I} P\left(Y_{d,i} = c_j\right) \cdot P\left(X_{d,k} > c_j, 0 \leq k < i\right) \\ = & \sum_{j=1}^{I} g_d\left(c_j\right) \cdot \left(1 - F_d\left(c_j\right)\right)^i \end{aligned} \tag{3}$$

With Equation 3 and using the geometric sum we are now able to derive the formula for the expected number of nodes satisfying the condition in Equation 2.

$$
\begin{aligned}
&\sum_{d=0}^{n} \mathbb{E}\left[\left|\{i \mid Y_{d,i} < \min_{0\le k<i}\{X_{d,k}\}\}\right|\right] \\
=&\sum_{d=0}^{n}\sum_{i=0}^{b^d-1} P\left(Y_{d,i} < \min_{0\le k<i}\{X_{d,k}\}\right) \\
=&\sum_{d=0}^{n}\sum_{i=0}^{b^d-1}\sum_{j=1}^{I} g_d\left(c_j\right)\cdot\left(1-F_d\left(c_j\right)\right)^i \\
=&\sum_{d=0}^{n}\sum_{j=1}^{I} g_d\left(c_j\right)\cdot\sum_{i=0}^{b^d-1}\left(1-F_d\left(c_j\right)\right)^i \\
=&\sum_{d=0}^{n}\sum_{j=1}^{I} g_d\left(c_j\right)\cdot\frac{1-\left(1-F_d\left(c_j\right)\right)^{b^d}}{F_d\left(c_j\right)} \\
=&\sum_{d=0}^{n}\sum_{j=1}^{I} g_d\left(c_j\right)\cdot\frac{1-\left(1-F_n\left(c_j\right)\right)^{b^n}}{F_d\left(c_j\right)}
\end{aligned}
\tag{4}
$$

Theorem 2. *The expected number of nodes visited by a depth-first search Branch-and-Bound algorithm is polynomial if* $g_d\left(c_i\right) \le c \cdot n^k \cdot F_d\left(c_i\right)$ *for* $1 \le i \le I$, $0 \le d \le n$, *where* c *and* k *are constants.*

Proof. Let $g_d\left(c_i\right) \le c \cdot n^k \cdot F_d\left(c_i\right)$, $1 \le i \le I$, $0 \le d \le n$.

$$
\begin{aligned}
\sum_{d=0}^{n} \mathbb{E}\left[\left|\{i \mid Y_{d,i} < \min_{0\le k<i}\{X_{d,k}\}\}\right|\right] &= \sum_{d=0}^{n}\sum_{i=1}^{I} g_d\left(c_i\right)\cdot\frac{1-\left(1-F_n\left(c_i\right)\right)^{b^n}}{F_d\left(c_i\right)} \\
&\le \sum_{d=0}^{n}\sum_{i=1}^{I} c\cdot n^k\cdot F_d(c_i)\cdot\frac{1-\left(1-F_n\left(c_i\right)\right)^{b^n}}{F_d\left(c_i\right)} \\
&\le \sum_{d=0}^{n}\sum_{i=1}^{I} c\cdot n^k = c\cdot I\cdot(n+1)\cdot n^k
\end{aligned}
$$

As I was assumed to be polynomial in n, $c \cdot I \cdot (n+1) \cdot n^k$ is a polynomial in n. □

Corollary 1. *The expected number of nodes visited by a depth-first search Branch-and-Bound algorithm with any lower bound function is polynomial if* $F_n\left(c_1\right) = \Omega\left(\frac{1}{n^k}\right)$, *where* k *is some constant.*

Proof. The worst case for the depth-first search Branch-and-Bound algorithm would be if $g_d(c_1) = 1$, i.e., the lower bound function always provides the smallest possible value c_1. In that case, the search would stop as soon as the first leaf node with a value of c_1 is found, otherwise if no such leaf node exists, the whole Branch-and-Bound tree is traversed. Therefore without loss of generality we can assume that $g_d(c_1) = 1$ and $g_d(c_i) = 0$ for $1 < i \leq I$. Also let $F_n(c_1) = \Omega\left(\frac{1}{n^k}\right)$.

$$\begin{aligned}
\sum_{d=0}^{n} \mathbb{E}\left[\left|\{i \mid Y_{d,i} < \min_{0\leq k<i}\{X_{d,k}\}\}\right|\right] &= \sum_{d=0}^{n}\sum_{i=1}^{I} g_d(c_i) \cdot \frac{1-(1-F_n(c_i))^{b^n}}{F_d(c_i)} \\
&\leq \sum_{d=0}^{n} \frac{1-(1-F_n(c_1))^{b^n}}{F_d(c_1)} \\
&\leq \sum_{d=0}^{n} \frac{1}{F_d(c_1)} = \sum_{d=0}^{n} \frac{1}{1-(1-F_n(c_1))^{b^{n-d}}} \\
&\leq (n+1) \cdot \frac{1}{F_n(c_1)} = \mathcal{O}(n^{k+1})
\end{aligned}$$

□