Game Tree Searching by Min/Max Approximation – Summary

This paper introduces the idea of approximating min and max operators through the use of generalised mean values, as opposed to using discrete values to propagate them. The generalised mean value is of the form:

$$M_p(a) = \left(\frac{1}{n}\sum\nolimits_{i=1}^{n} a_i^p\right)^{1/p}$$

Where n: number of nodes (i.e. branching factor)
a_i: min / max values of children nodes n
p ∈ $\mathbb{R} \setminus \{0\}$

$M_p(a)$ gives the approximate max/min value, which quickly converges to the actual max/min values as $\lim_{p \to \pm\infty}$. The advantage is that this function is continuous and differentiable everywhere. This allows the computation of partial derivatives of the root node with respect to any of its successors. Being able to compute partial derivatives implies knowledge over which of the leaf nodes have the most impact in the backed-up value of the root node. Having this information allows to explore the tree asymmetrically, in decreasing order of the values of partial derivatives of the root node with respect to its children. In other words, it first explores the children nodes that carry the highest degree of volatility in the root node. This appears to be similar to quiescence search – while quiescence search prefers depth-first search until variance between levels falls below some threshold, min/max approximation prefers a more thorough search of branches that are most significant in relation to the value of the root.

This approach leads to worse performances compared to alpha-beta pruning. Due to the additional strain of computing partial derivatives, it explores a significantly smaller portion of the game tree than alpha-beta in the same time period. However, if instead of a time restriction we employ a restriction in the size of the sub-tree we're allowed to search, it is possible that min/max approximation will outperform alpha-beta due to the selective process by which it explores the tree.
This approach seems more similar to the strategies humans employ in game playing. It seems reasonable that a Chess player prefers some moves over others and he does not explore all options uniformly – this leads to some means of sorting the moves and allocating time to explore them accordingly.

If I understand correctly, this method allows for an asymmetric exploration of the game tree as opposed to the exploration that results from MM with α-β pruning. However, it doesn't seem to be *a priori* clear that min/max approximation leads to a drastically different pattern of game tree exploration. It seems intuitive that if we consistently use the approach of first exploring the leaf node with maximal partial derivative with respect to the root node, the outcome of the exploration will not differ greatly from MM with α-β pruning. This, I believe, is because as we delve deeper into any given branch of the tree, the influence of the leaf node at depth n on the root node decays rapidly, and it is exceeded by the partial derivatives of nodes closer to the root. I couldn't find a reason why the partial derivatives wouldn't be dependant on the length between nodes. In other words, I find it unlikely that based on this approach alone we'll get significant enough deviations from uniformity in order to account for the added strain of computing partial derivatives. Nevertheless, having a differentiable max function seems useful.