Review of Game Tree Searching by Min/Max Approximation

A brief summary of the paper's goals or techniques introduced.

The goal of the paper is to present an iterative method for searching minlmax game trees. based on the idea of approximating the “min” and “max” operators by using a generalised mean-valued operators.
The idea is to expand the nodes that have the largest effect on the current node value. By using the generalised mean to approximate min max values. The min or max of a set of numbers is defined through a function. You can obtain a continuous derivative with respect to all arguments. By Obtaining this partial derivate you can target the child node which has the greatest affect on the parent. This is the node that is to be explored.

Resource Bound per turn	Minimax aprox wins	Alphabeta wins	Ties
1 second	41	46	11
2 seconds	*40*	42	16
3 seconds	36	44	18
4 seconds	39	52	7
5 seconds	30	55	13
TOTAL	186	239	65
1000 moves	47	35	16
2000 moves	50	35	13
3000 moves	42	47	9
4000 moves	49	42	7
5000 moves	61	31	6
TOTAL	249	190	51

A brief summary of the paper's results (if any).

An agent using the minimax approximation algorithm played 980 games against an opposition using the alphabeta algorithm. The games covered all possible starting positions for both agents. The results of the game can be seen above.
Two resource bounds are used to evaluate the effectiveness of the algorithm. One being number of ‘moves’ which is the number of nodes traversed. The second being the CPU time used. The bounds being per turn, in the experiment the bounds ranged from 1 to 5 seconds (in 1 second increments) and 1000 to 5000 moves (in 1000 move increments). From the table above it can be seen that the alphabeta agent came out superior when moves were time limited. The alternative can be said when looking at the number of bound moves allowed.