Potential applications of opponent-model search (Part 1

Iida, H.; Uiterwijk, J.W.H.M.; van den Herik, H.J.; Herschberg, I.S.

Published in:
ICCA Journal

Publication date:
1993

Citation for published version (APA):
Iida, H., Uiterwijk, J. W. H. M., van den Herik, H. J., & Herschberg, I. S. (1993). Potential applications of opponent-model search (Part 1: The domain of applicability). *ICCA Journal*, *16*(4), 201-208.

POTENTIAL APPLICATIONS OF OPPONENT-MODEL SEARCH[1]

Part 1: The Domain of Applicability

Hiroyuki Iida[2], *Jos W.H.M. Uiterwijk and H.J. van den Herik*

Department of Computer Science
University of Limburg
Maastricht, The Netherlands

I.S. Herschberg

Department of Technical
Mathematics and Informatics
Delft University of Technology
Delft, The Netherlands

ABSTRACT

An opponent is modelled by assumed knowledge of his evaluation of positions in a game. Exploiting this knowledge and assuming the opponent to be fallible, the opponent may be outwitted by anticipating his errors. Though the moves so generated need not be optimal in some minimax sense, the model may confer an advantage to the modelling player. Conditions are derived for what is, in essence, a minimum distance between the two player's strategies; notably, an impetuous opponent is seen to labour under the same disadvantage as one with shallower search depth.

1. INTRODUCTION

At present, most search algorithms in game-playing programs are still based on minimax theory, published more than half a century ago by von Neumann (1928): they usually incorporate the α-β refinement of Knuth and Moore (1975). This combination, it is known, may fail to yield the best results. Accordingly, several other non-minimax search strategies have been investigated, e.g., those based on using several best- or worst-successor nodes (Slagle and Dixon, 1970), or even accounting probabilistically for all successor nodes (Ballard, 1983; Reibman and Ballard, 1983). These, due to their inefficiency, are not competitive with the α-β algorithm.

More recently, Uiterwijk and Van den Herik (1994) investigated speculative play: the α-β algorithm was enhanced by endowing it with bonus/malus terms favouring propitious moves and/or those threatening the opponent even when they depart slightly from the minimax optimum. Jansen (1992, 1993) more deeply investigated speculative play for a subdomain of chess, the KQKR endgame. He provided evidence that human opponents playing the stronger side may be badly hampered in their progress to their game-theoretical victory, or may even be forced into a draw by a judicious choice of moves by the program playing the weaker side. The latter choice is guided by perfect database knowledge of the domain combined with awareness of common human heuristics often applied in this endgame. In this sense, Jansen may have been the first author to incorporate a model of the opponent in a program.

This idea of anticipating the opponent's expected strategy is certainly not new in the human arena. Consider any strategic game played by two grandmasters. Other things being equal, they will use very similar evaluation functions at a very similar search depth. This is confirmed by the senior author's experience as a grandmaster in Shogi.

---

[1] This is part 1 of a revised version of the paper delivered under the title *Thoughts on the Application of Opponent-Model Search* on July 2, 1993, at the 7[th] Conference on Advances in Computer Chess, Maastricht, the Netherlands. It will be published in full in the Proceedings of that conference.

[2] Presently affiliated to the Department of Computer Science, Tokyo University of Agriculture and Technology.

As a working hypothesis, we advance that one of the factors deciding the outcome of the game is strategic. As to the content of the strategy, we take a hint from experiments analysing the thinking processes of Shogi grandmasters (Iida and Kotani, 1992), where mostly the players attempted to understand the intention behind the opponent's previous moves. In an equivalent formulation, their choice of move was in part decided by their interpretation of the opponent's strategy. Rephrasing this again: understanding the opponent by modelling him would, on this hypothesis, allow them to obtain a better result than would be obtainable without such a model.

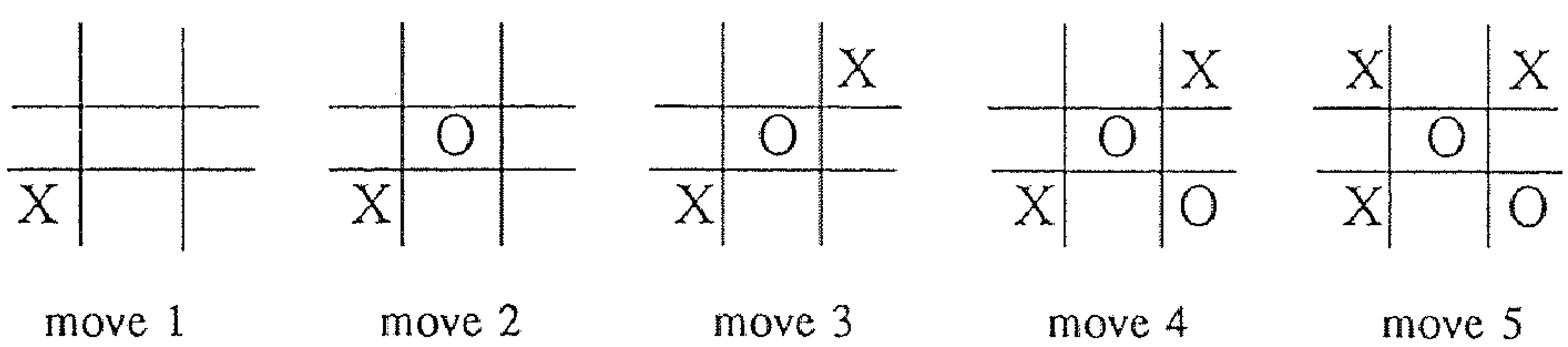
Granted we have knowledge of the opponent's evaluation function and assuming, moreover, that the opponent is modelled by this function to a sufficient approximation, a search based on such an opponent model, OM-search for short, can be formalized as a search technique with the potential of reaching a position that may be better, but should never be worse, than the one dictated by the minimax strategy (Iida and Uiterwijk, 1992).

As a simple example (from Van den Herik, 1988), consider playing TicTacToe by the following strategic rules **R** and the heuristics stated **H**, to be applied in the order given.

> $\mathbf{R}_1$: if completing three-in-a-row is possible, do so.
> $\mathbf{R}_2$: if the opponent threatens completing three-in-a-row, prevent this if possible.
> $\mathbf{H}_1$: occupy the central square, whenever possible.
> $\mathbf{H}_2$: occupy a corner square, whenever possible.

$\mathbf{H}_1$ and $\mathbf{H}_2$ have been formulated as heuristics rather than rules since they directly translate the consideration governing them, viz. that the central square is most important of all, and that among the border squares, those at the corners are more important than the others.

TicTacToe is known to be drawn, and it might be questioned whether knowledge of one's opponent strategy could improve on this result. Intuitively, it seems clear that S, by definition being $\{\mathbf{R}_1, \mathbf{R}_2, \mathbf{H}_1, \mathbf{H}_2\}$, should achieve a draw, since it correctly evaluates the importance of the squares and acts on this evaluation. Yet, a program aware of the opponent's strategy S may win. Allow the program the first move as X, the following sequence of moves then causes X to win, where at move 2 and 4 O follows S.



The win by X is due to X's awareness of the opponent's strategy S, admittedly non-optimal, or to rephrase this statement, due to X's successful prediction of O's moves. X may be truthfully stated to apply OM-search in this simplest of examples.

2. OPPONENT-MODEL SEARCH

In this section, the algorithm of the OM-search is sketched and its most important characteristic is stated, viz. that it can never yield worse results than minimax search. An example is supplied. For enhancements of the implementation due to refinements in pruning see the internal report by Iida, Uiterwijk and Van den Herik (1993).

2.1 Formalization of the Idea

OM-search is predicated on the assumption that one has complete knowledge of the opponent's strategy and that this strategy is completely contained in the opponent's evaluation function. Although this may seem extremely restrictive, it is no more restrictive than conventional search techniques which are based on the same assumptions, specifically that the opponent's evaluation function coincides with the one applied by the player.

In OM-search, *two* values are computed for all positions in a game tree. For clarity, the players are distinguished as a max player and a min player. The algorithm is indicated in a coarse-grained manner by (2.1) and (2.2) below according to (Iida *et al.*, 1993); the same formulation had been arrived at before by Carmel and Markovitch (1993).

Let i, j range over all immediate successor positions of the node considered. Let a max-node be defined as a node where the max player is to move, and analogously for a min-node. Let the functions *ma* and *mi* be defined for all positions, where *ma* is an evaluation function as adopted by the max player and *mi* similarly for the min player. It further is assumed that the opponent (min) player always uses some minimax response strategy.

Then

$$ma(P) = \begin{cases} \max_i ma(P_i) & \text{if } P \text{ is a max node} \\ ma(P_j) \text{ with } j \text{ such that } mi(P_j) = \min_i mi(P_i) & \text{if } P \text{ is a min node} \\ EV_{ma}(P) & \text{if } P \text{ is a terminal node} \end{cases} \qquad (2.1)$$

$$mi(P) = \begin{cases} \max_i mi(P_i) & \text{if } P \text{ is a max node} \\ \min_i mi(P_i) & \text{if } P \text{ is a min node} \\ EV_{mi}(P) & \text{if } P \text{ is a terminal node} \end{cases} \qquad (2.2)$$

$EV_{ma}(P)$, defined only for terminal nodes, denotes the value at node P of the max player's evaluation function and is necessarily static. $EV_{mi}(P)$ analogously denotes the value for a terminal node of the min player's evaluation function.

Let us show, in Figure 1, an example of OM-search and compare it with the minimax strategy.

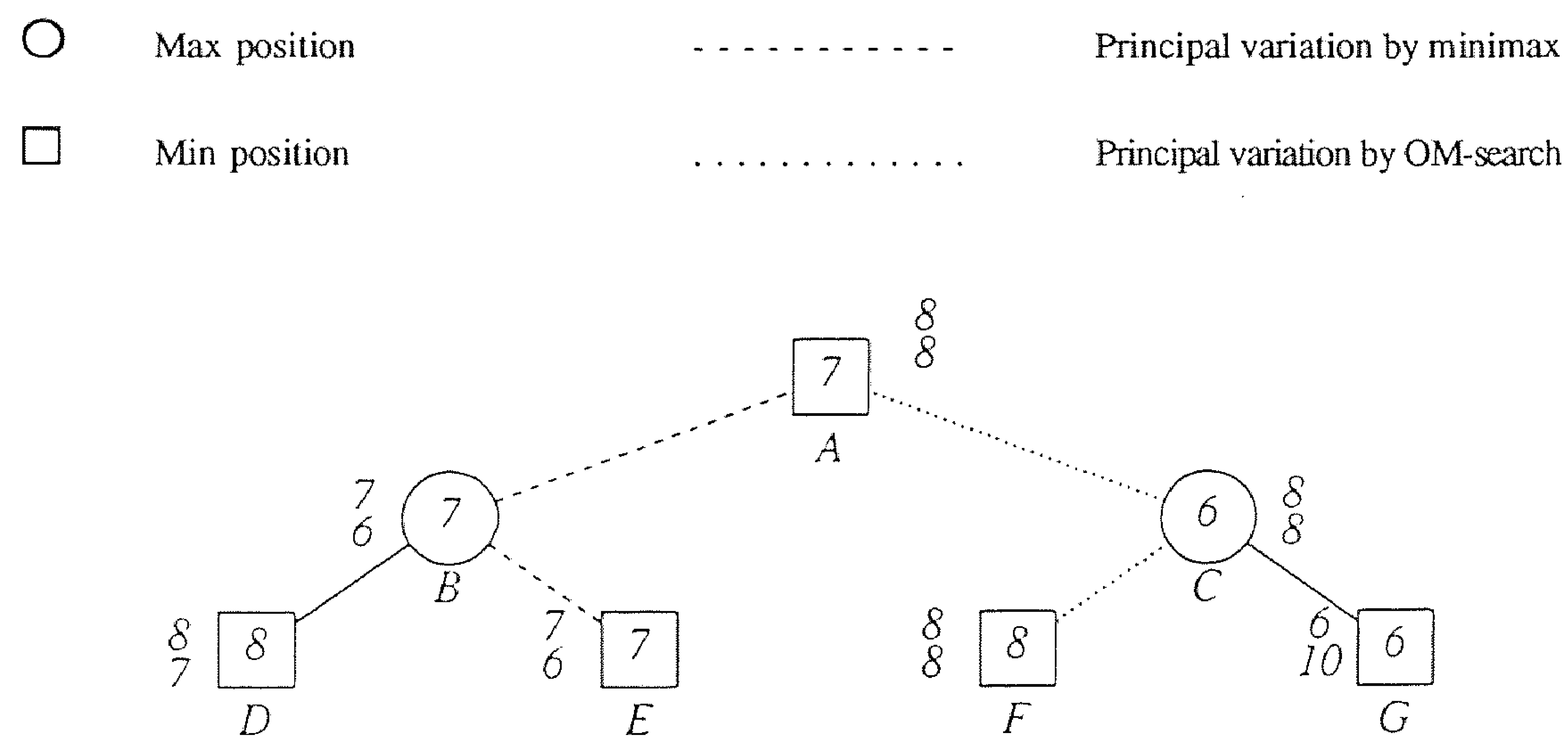


Figure 1: The OM-search and the minimax strategy compared.
The numbers *beside* the nodes represent the values by OM-search, the upper for the max player's evaluation function and the lower for the min player's dito. The number *within* each node represents the minimax value for that position.

In this game tree, the root values obtained by OM-search and minimax differ because the players have adopted differing evaluation functions. In this example, the max player expects the min player to make a serious mistake in the right-hand branch of the tree, viz. considerably overestimating the value of node G. Anticipating this, the max player will get a better result by opting in the root position for the move corresponding to the right-hand node, in spite of that move being non-optimal in the minimax sense.

2.2 A Characteristic of OM-Search

An important characteristic of OM-search directly follows from the theorem below, a proof of which is in Iida *et al.* (1993).

Theorem 1

OM-search of a game tree by the max player will have a value not less than the value derived by some minimax strategy.

As an implied condition on the theorem, the opponent should behave as predicated by his assumed evaluation function. A wider consequence of the theorem is that if the opponent makes an error (according to the max player's evaluation function) OM_search will allow exploitation of that error. This is a characteristic of OM-search.

3. APPLYING OM-SEARCH

In this section, we describe an analysis of thinking times in a selection of games played by Shogi top grandmasters. Some reasonable strategies derived from the analysis will be defined and some characteristics of applying OM-search will be shown.

In the following discussion, we assume that both players use the same evaluation function; this implies that the model of the opponent only differs from that of the player to move in the depth to which the opponent is assumed to look ahead and in his response strategy.

3.1 An Analysis of Thinking Times in Shogi Games

We analyzed the distribution of the response times as observed for eleven Shogi top grandmasters (see the Appendix). In the analysis, we observed that their response often is immediate. Let it be assumed that their immediate response arose from selecting a move which they had precalculated to be best when considering their previous move or an even earlier one. If true, this implies that an immediate response is based on a search which is shallower by at least two plies than a more deliberate response. This train of reasoning obviously ignores any effect arising from the fact that the opponent's thinking presumably continues during *his* opponent's thinking time.

In order to characterize an opponent's possible behaviours, let us attach some verbal labels to observed respons strategies. For a start, we relate them to a presumed depth of search.

Definition 1 [*response strategies*]

Let d_X be the depth in plies to which some player X is presumed to look ahead in his average thinking time. We define the **normal** response strategy by X as a search to depth d_X.
We identify four other response strategies with their presumed search depths as follows

immediate	d_X-2
quick	d_X-1
normal	d_X
slow	d_X+1
very slow	d_X+2

The identification above is based on the assumption that the immediate strategy has been immediately preceded only by a search with the normal strategy (i.e., at the previous move). Again, the effects of thinking on the opponent's time, it must be pointed out, have been ignored.

In the above, we have linked the response time of X to his depth of search. It is equally plausible to link his habitual response time to his other personal characteristics as observable in the timing of his responses. Rather arbitrarily, let us assign the values 2, 1, 0, −1 and −2 to the very slow, slow, normal, quick and immediate response strategies, respectively, indicating the difference in plies from the average search depth of player X.

Still, a player sensing a difficulty ahead may opt to depart from (in effect, prolong) his habitual response time as the occasion demands. Prolonging the response time in such cases will impact the depth of search.

3.2 When to Apply OM-Search

For definiteness' sake, let us posit in the ensuing text that player A follows the OM-search whenever applicable, and that player B consistently applies some minimax-based strategy.

Definition 2 [*depth difference*]

The **depth difference** δ is defined as the difference between the average look-ahead depths (in normal thinking time) of players A and B. Let d_A and d_B be the values for these average depths for player A and B. We then define the **depth difference** as $\delta = d_A$-d_B.

Definition 3 [*response-time difference*]

The **response-time** difference τ is defined as the difference in the response strategies used by the players A and B. Let t_A and t_B be the values for the response strategies used by player A and B. We then define the **response-time difference** as $\tau = t_A$-t_B.

Lemma 1

Let δ be the depth difference and τ be the response-time difference as defined above. If $\delta + \tau \geq 2$, then OM-search can be applied.

Proof

Let d_A be player A's average look-ahead depth and let his particular response strategy for the move under consideration be indicated by the value t_A. Similarly suppose that player B's look-ahead depth and response strategy are given by d_B and t_B. In a position with player A to move, A will look ahead (d_A+t_A) plies. Player A expects that player B will look ahead (d_B+t_B) plies at his following move, corresponding with a depth of (d_B+t_B+1) plies from the position from which A is to move. Using $d_B = d_A-\delta$ and $t_B = t_A-\tau$ this equals d_A+t_A+1 − $(\delta+\tau)$ plies. If $\delta + \tau \geq 2$, player A knows that B consequently will not look ahead to the leaf nodes at (d_A+t_A) plies. Therefore, at level (d_B+t_B+1) plies two (possibly different) values for each node are available, one stemming from player B's static evaluation and one being a back-up value (in the minimax sense) for the search trees at level (d_B+t_B+1) extending to (d_A+t_A) plies. Hence, OM-search can be applied to the (d_B+t_B+1) tree.

Corollary:

The condition $\delta + \tau \geq 2$ defines the minimum difference in combined depths and response times at which OM-search is applicable.

Definition 4 [*player's model*]

For a player X with average look-ahead depth d_X and expected response-time strategy t_X, we define a player's **model** m_X as the sum: $m_X = d_X+t_X$, for definiteness $m_X > 0$.

For any position P and any player X, the following values V are well defined.

$V(P, mm, m_X)$,	the value of P obtained by some minimax strategy mm, using model m_X.
$V(P, om, m_X, m_Y)$,	the value of P obtained by OM-search om, using m_X and m_Y as the respective players' models.

By theorem 1 we have

$$V(P, om, m_X, m_Y) \geq V(P, mm, m_X) \qquad (3.1)$$

It is expedient to distinguish in (3.1) those cases where strict inequality holds. If it does not, opponent's errors may have been detected which, however, do not impact $V(P, om, m_X, m_Y)$. Whenever they do so, the more interesting case of strict inequality $V(P, om, m_X, m_Y) \geq V(P, mm, m_X)$ arises, which we shall examplify, for $\delta + \tau = 2$, the simplest case, in Figure 2.

Let player X perform an OM-search with m_X=4 and m_Y=2; similarly assume that Y applies some minimax search at 2-ply depth. Studying the tree and on the models employed, we find that at A, the root node, $V(P, om, m_X, m_Y)$=11 > $V(P, mm, m_X)$=10. The discrepancy arises because in the OM-search, an error by Y is anticipated which impacts the value of the root. The error mainly originates in node J and is propagated upward through E to B, where Y as a consequence will take the wrong continuation.

○	Max position	- - - - - - - - - - -	Principal variation by minimax
□	Min position	. . . .	Principal variation by OM-search

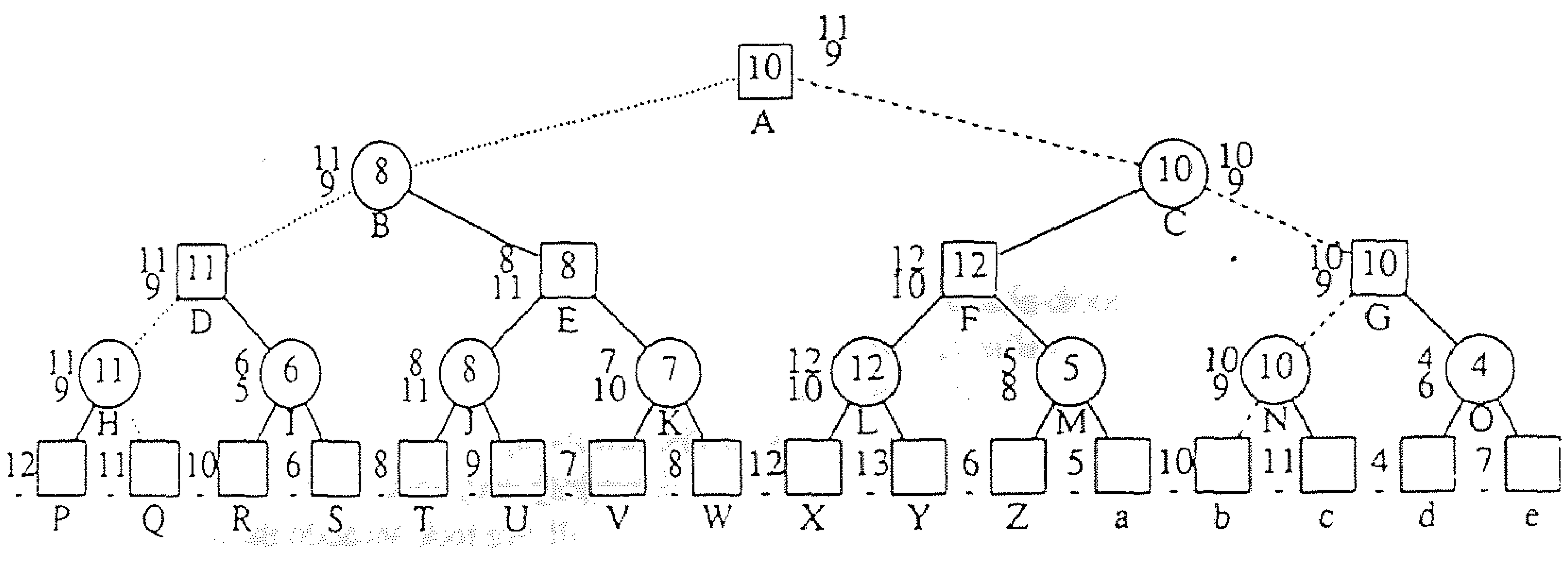


Figure 2: An example of applying OM-search with m_X = 4 and m_Y = 2.
The numbers *beside* the nodes represent the values by OM-search, the upper for the max player's evaluation function (applied at depth 4) and the lower for the min player's ditto (applied at depth 3). The number *within* each node represents the minimax value for that position (using the evaluation values obtained at depth 4).

3.3 In How Many Ways Can OM-Search Be Applied?

Each way of applying OM-search is completely defined by m_X and m_Y, where, for definiteness, $m_X \geq m_Y + 2$ (from lemma 1 and definition 4). By simple discrete summation, we find for the number of ways, considering that player Y may, from instance to instance, choose any model with depth at most equal to m_Y and since player X may respond by choosing his m_X to match, that

$$N(m_X, m_Y) = \sum_{i=1}^{m_Y} (m_X - i - 1) = m_X \times m_Y - {}^1/_2\, m_Y\, (m_Y + 3)$$

4. ANTICIPATORY NOTE

Having probed the domain of applicability of the opponent's model search, we shall devote Part 2 of this paper to a more detailed consideration of the risks the opponent's model may carry and how to control them. Doing so, a choice of strategies will be (non-exhaustively) indicated, all designed to minimize the associated risk in one of several senses. A final example will be presented in which the risk-avoiding strategies will all differ one from another.

5. REFERENCES

Ballard, B.W. (1983). The *-Minimax Search Procedure for Trees Containing Chance Nodes. *Artificial Intelligence*, Vol. 21, pp. 327-350.

Carmel, D. and Markovitch, S. (1993). Learning Models of Opponent's Strategy in Game Playing. *CIS Report* #9305, Technion - Israel Institute of Technology, Haifa, Israel. A slightly different publication under the same title has appeared as *AAAI Technical Reports* #FS93-02: Games & Learning.

Herik, H.J. van den (1988). *Informatica en het Menselijk Blikveld*. Inaugural address. University of Limburg, Maastricht. (In Dutch) ISBN 90-72710-01-0.

Iida, H. and Kotani, Y. (1992). A Strategy of Game Tree Search Modelling Experts' Thinking Process. *The Transactions of Information Processing of Japan*, Vol. 33, No. 11, pp. 1296-1305. (In Japanese)

Iida, H. and Uiterwijk, J.W.H.M. (1992). How to Become a Shogi Grandmaster. *Proceedings of The First European Shogi Workshop*. Shogi Deutschland and EMBL, Heidelberg, pp. 25-29.

Iida, H., Uiterwijk, J.W.H.M. and Herik, H.J. van den (1993). Opponent-Model Search. *Technical Reports in Computer Science*, Dept. of Computer Science, University of Limburg, Maastricht.

Jansen, P.J. (1992). *Using Knowledge about the Opponent in Game-Tree Search*. Ph.D. thesis, Carnegie-Mellon University, Pittsburgh, PA.

Jansen, P.J. (1993). KQKR: Speculatively Thwarting a Human Opponent. *ICCA Journal*, Vol. 16, No. 1, pp. 3-17.

Knuth, D. E. and Moore, R.W. (1975). An Analysis of Alpha-Beta Pruning. *Artificial Intelligence*, Vol. 6, No. 4, pp. 293-326.

Neumann, J. von (1928). Zur Theorie der Gesellschaftsspiele. *Math. Ann.*, Vol. 100, pp. 295-320. Reprinted (1963) in *John von Neumann Collected Works* (ed. A.H. Taub), Vol. VI, pp. 1-26. Pergamon Press, Oxford.

Reibman, A.L. and Ballard, B.W. (1983). Non-Minimax Search Strategies for Use against Fallible Opponents. *Proc. Third national Conf. Artificial Intelligence AAAI'83*, pp. 338-342.

Slagle, J.R. and Dixon, J.K. (1970). Experiments with the M & N Tree-Searching Program. *Comm. ACM*, Vol. 13, No. 3, pp. 147-154.

Uiterwijk, J.W.H.M. and Herik, H.J. van den (1994). Speculative Play in Computer Chess. To appear in *Advances in Computer Chess 7* (eds. H.J. van den Herik, I.S. Herschberg and J.W.H.M. Uiterwijk).

Appendix: ANALYSIS OF THINKING TIMES

The statistics of 10 Shogi games were gathered and summarized in Table 1. They all have been played in 1992. TIME is the total time for each player in each game, while MOVES gives the length of the games (summing White's and Black's moves). IRM, QRM, NRM, SRM and LRM denote the number of moves for which the thinking time is less than IRMT, greater than IRMT and less than QRMT, higher than QRMT and less than SRMT, higher than SRMT and less than LRMT, and higher than LRMT, respectively (For closer definition, see below). The ratio of each type of moves is given between brackets.

White vs. Black	TIME	MOVES	IRM	QRM	NRM	SRM	LRM
1. Murayama vs. Tanigawa	8 hours	126	60 (48%)	11 (9%)	13 (10%)	15 (12%)	27 (21%)
2. Tanigawa vs. Habu	4	103	74 (72)	11 (11)	8 (8)	3 (3)	7 (7)
3. Minami vs. Tanaka	6	162	75 (46)	19 (12)	18 (11)	12 (7)	38 (23)
4. Kobayashi vs. Tanigawa	6	117	57 (49)	9 (8)	10 (9)	18 (15)	23 (20)
5. Hiura vs. Yashiki	6	127	57 (45)	10 (8)	8 (6)	22 (17)	30 (24)
6. Tanigawa vs. Murayama	8	97	53 (55)	4 (4)	15 (15)	2 (2)	23 (24)
7. Murayama vs. Sakurai	6	87	50 (57)	5 (6)	9 (10)	8 (9)	15 (17)
8. Morishita vs. Kodama	6	99	56 (57)	8 (8)	10 (10)	7 (7)	18 (18)
9. Tanigawa vs. Habu	5	97	46 (47)	11 (11)	14 (14)	8 (8)	18 (19)
10. Murayama vs. Tanigawa	8	94	55 (59)	1 (1)	13 (14)	3 (3)	22 (23)

Table 1: An analysis of thinking times in Shogi games played by top grandmasters.

The classification of moves given in the table is done using the following parameters. B represents the branching factor. ATEM is calculated as ATEM = TIME/MOVES, which corresponds to the average thinking time for each move. IRMT is a value calculated as IRMT = ATEM/B, which corresponds with searching two plies less deep than on average (the criterion for an immediate-response move). Similarly, QRMT, SRMT and LRMT are calculated as $ATEM/\sqrt{B}$, $ATEM \times \sqrt{B}$ and $ATEM \times B$, and correspond to the criteria for quick (one ply less deep), slow (one ply deeper) and very slow (two ply deeper) response moves.

Notice that in this analysis we assumed the branching factor for top Shogi grandmasters to be 2.5, which is justified by the senior author's experience as a Shogi grandmaster. Further it is assumed that they also use the α-β technique, and we calculated each time rounded to the nearest integer from each game score.