AEL Pruning


AEL PRUNING¹

Ernst A. Heinz²

Cambridge, USA

ABSTRACT

In the past, we introduced the new forward-pruning techniques adaptive null-move pruning (A), extended futility pruning (E), and limited razoring (L). Since they differ substantially in nature, it remained unclear how well they combine and which level of effectiveness their combination achieves. Based on the names of the three building blocks, we call the combined scheme AEL pruning. As it turns out, AEL pruning clearly belongs to the collection of practically successful forward-pruning techniques in computer chess.

AEL pruning is easy and efficient to implement. Extensive experiments with 2180 test positions from well-known tactical test suites show that AEL pruning fully preserves the tactical strength of our chess program DARKTHOUGHT while reducing its search effort by 20 to 50 percent on average at fixed iteration depths of 8 to 12 plies. The reduction in search effort scales as well with search depth as that for any of the three techniques alone. The positive results of 580 test games (self-play and versus other strong chess programs) provide further empirical evidence for the practical usefulness of AEL pruning.

1. INTRODUCTION

Our recent articles (Heinz, 1998, 1999) elaborated on the three new forward-pruning schemes adaptive null-move pruning, extended futility pruning, and limited razoring. We specifically described how the novel techniques function and how well they work on their own. While extended futility pruning and limited razoring are static forward-pruning schemes, adaptive null-move pruning is a dynamic one. In order to quantify their individual performances, we intentionally restricted our prior experiments to test suites because they allowed us to assess the practical viability and tactical safety of each single scheme. Following up thereon, we now extend our focus and scope of validation in two different ways.

1. We combine adaptive null-move pruning (A), extended futility pruning (E), and limited razoring (L). Section 3 introduces the resulting combined scheme which we call AEL pruning.

2. In addition to our experiments with 2180 positions from tactical test suites (see Section 3.2), we executed 300 self-play games and 280 test games versus other strong chess programs (see Section 4) with the AEL-pruning version of our tournament-proven chess program DARKTHOUGHT¹ (Heinz, 1997).

Unfortunately, there exists no theory of how to best evaluate forward-pruning schemes in game-tree search. Practical experience recommends test suites as adequate tools for the quantification of changes in tree size and raw tactical strength. Complementing these objectives, test games measure the overall playing capabilities of forward-pruning searchers as resulting from their positional and tactical skills combined. Hence, test games and test suites measure different characteristics of the programs in question. Both seem to be necessary in order to gain an adequate understanding of the overall effects caused by any search or evaluation refinements.

¹ This article is an extended version of the identically titled Chapter 3 from our book "Scalable Search in Computer Chess" (Heinz, 2000). We thank the publisher Vieweg Verlag for their kind permission to reprint large parts of the chapter here.
² M.I.T. Laboratory for Computer Science (Room NE 430-228), Massachusetts Institute of Technology, 545 Technology Square, Cambridge, MA 02139, USA. E-mail: heinz@mit.edu. WWW = http://homepages.mit.edu/heinz/
³ WWW = http://homepages.mit.edu/heinz/
2. RELATED WORK

Our previous articles (Heinz, 1998, 1999) already discussed related research that we deem relevant for our current work. Hence, we refer to the sections on related work of these articles and refrain from reiterating them in detail again. For the sake of completeness, we only provide a keyword list of the corresponding citations here: computer chess and search up to late 1980s – Marnland (1992), razing – Birmingham and Kent (1977), GAMMA algorithm – Newborn (1975), marginal forward pruning – Slagle (1971), futility pruning – Saito and Akira (1977), Schaeffer (1986), Ye and Marnland (1992), null move – Beal (1989, 1990), Goertzel and Campbell (1990), Donninger (1993), multi-cut alpha-beta pruning – Bjornsson and Marnland (1998, 1999), ProBcUT and Multi-ProBcUT – Buro (1995, 1997).

However, we briefly review and summarize our prior findings about each of the AEL components. We think this is appropriate in order to achieve a better understanding of the fully combined pruning scheme.

Adaptive Null-Move Pruning (A)

Standard recursive null-move pruning relies on a constant depth-reduction factor $R = 2$. Independent studies by several researchers confirmed that $R = 2$ behaved markedly better than both the too conservative $R = 1$ and the too aggressive $R = 3$. In view of this we considered adaptive depth-reduction factors instead of fixed ones and devised the $R = 3^n/2$ formula (Heinz, 1999). $R = 3^n/2$ combines the merits of both $R = 2$ (tactical safety) and $R = 3$ (reduced search effort). We assessed the practical viability of adaptive null-move pruning by letting Dark Thought with $R = 2$ and $R = 3^n/2$ search all 2180 positions of the well-known tactical test suite Encyclopedia of Chess Middlegames (ECM), Win at Chess (WAC), and 1001 Winning Chess Sacrifices (WCS) to fixed iteration depths of 8, 10, and 12 plies respectively. The experiments showed that adaptive null-move pruning performs equally well at tactics as standard null-move pruning with $R = 2$ while reducing the search effort by 10 to 30 percent on average at search depths of 8 to 12 plies. Moreover, the observed reduction in search effort scales nicely with increasing search depth.

Extended Futility Pruning (E)

The main innovation of extended futility pruning is to apply the futility idea at pre-frontier nodes with a remaining search depth of 2 plies (Heinz, 1998). It cuts complete branches of the search tree (quiet moves and non-checking captures) which fulfill the purely static, extended futility condition. Although extended futility pruning is theoretically unsound, it works very well in practice. Slight yet important modifications and additions with respect to normal futility pruning achieve the proper safety precautions that greatly reduce the selective risks of the extended scheme. Our experiments with all 2180 positions from ECM, WAC, and WCS provided convincing empirical evidence for the benefits of the new technique. While scaling nicely with increasing search depth and hardly compromising any tactical strength, extended futility pruning reduced the search trees of Dark Thought by 12 to 21 percent on average as compared with normal futility pruning at search depths of 8 to 12 plies.

Limited Razing (L)

Pushing the futility idea one step further, limited razing applies it at pre-pre-frontier nodes with a remaining search depth of 3 plies (Heinz, 1998). The overall effect of limited razing is to cut quiet moves and non-checking captures with little material gains while reducing the remaining depth of all other moves by 1 ply at pre-pre-frontier nodes for which the completely static, limited razer condition holds. Although limited razing looks even riskier than extended futility pruning, it works equally well in practice. Our experiments with all 2180 positions from ECM, WAC, and WCS provided significant empirical evidence for the benefits of the technique, especially at search depths of more than 10 plies. The data showed that limited razing did not hamper the tactical abilities of Dark Thought at all while shrinking its search trees by an additional 8 percent on average as compared with extended futility pruning at a search depth of 12 plies.
3. COMBINED AEL PRUNING

The combination of adaptive null-move pruning, extended futility pruning, and limited ramoning must not only ensure the correct interplay of the techniques with each other but also with the rest of the search. We explain the theory, i.e., the right sequence of actions and the necessary safety precautions, in Section 3.1. Then, we validate the practical usefulness of AEL pruning by qualitative and quantitative experiments with thousands of positions from well-known test suites in Section 3.2 and by reporting on 580 full test games in Section 4.

3.1 Theory

Merging the listings from Figure 2 of Heinz (1999) and Figure 7 of Heinz (1998), we illustrate the crucial algorithmic aspects of AEL pruning in Figure 1. We present the skeleton of a selective search function that outlines the implementation of AEL pruning as executed on top of normal futility pruning. Of course, all remarks and explanations in Heinz (1998, 1999) regarding the implementation of adaptive null-move pruning, extended futility pruning, and limited ramoning also apply to AEL pruning. We refrain from repeating the respective details here once more. We refer the interested reader to the previous articles and to Heinz (2000).

The algorithmic description of AEL pruning in Figure 1 proves that it is easy and efficient to implement. The search skeleton exports the three futility margins as introduced in Heinz (1998). The safety precautions of extended futility pruning and limited ramoning are taken into account, too. AEL pruning performs adaptive null-move searches with a minimal window centered around the value of beta. But it tries the null-move search only if they really promise to cut off as decided by the condition ! prune and the function call to try-null(). Because null moves do not make much sense if the safe-to-move is a check or the opponent executed a null move directly before, we further guard the null-move search by the predicates !check (move) and null-okay(). The recursive call of the null-move search itself hides the exact formula of the adaptive depth-reduction factor inside the function/macro Redoct(). Last but not least, please note that the probe of the transposition table must follow after the potential depth reduction of limited ramoning.

3.2 Practice

We let DARKTHOUGHT with AEL pruning search all 2180 positions from the tactical test suites ECM, WAC, and WCS to fixed iteration depths of 8, 10, and 12 plies respectively. Table 1 lists the results of these searches and compares them with the results of the normal DARKTHOUGHT which employed only normal futility pruning and standard recursive null-move pruning with a constant depth-reduction factor $H = 2$. The data shows that AEL pruning compromises nearly no tactical strength while reducing the overall search effort by 20 to 50 percent on average at fixed depths of 8 to 12 plies.

The individual savings of adaptive null-move pruning, extended futility pruning, and limited ramoning add up almost completely when compared with the experimental data published in Heinz (1998, 1999, 2000). Simultaneously, the reduction in search effort scales at least as nicely with search depth as for any of the constituent techniques alone. The savings of AEL pruning scaled even better than linearly with progressing depth according to the numbers from Table 1. This further indicates how well adaptive null-move pruning, extended futility pruning, and limited ramoning combine with one another.

In addition to the above, the preserved tactical strength of DARKTHOUGHT with AEL pruning is also visible from the speeds, the solution rates, and the times to solution which it achieves for widely used test suites as presented in Table 2. There, "Time" means search time per position. Appendix C of Heinz (2000) gives the exact solution times for all positions from BS-2830, BT-2630, and LCT-II DARKTHOUGHT on a 500 MHz Compaq Alpha-21264 XP 1000 workstation for these test suite measurements. An identical version of DARKTHOUGHT (running on exactly the same machine) successfully competed in the 9th World Computer-Chess Championship in June 1999, earning the World Microcomputer-Chess Vice-Champion title and finishing on a shared 5th place as 6th of 30 participants overall.
int search(int alpha, int beta, int move, node parent, int depth) {
    node current;
    int extend, fmax, fscore, tt_hit;
    tt_entry tt_ref;
    /* declare the local variables that require constant initialization */
    int fprune = 0;
    int fpruned_moves = 0;
    int score = -infinite_val;
    /* execute the opponent's move and determine how to extend the search */
    make_move(parent, move, &current);
    extend = extensions(move, current, depth);
    depth += extend;
    /* decide about limited razoring at pre-pre-frontier nodes */
    fscore = (mat_balance(current) + razor_margin);
    if (!extend && (depth == pre_pre_frontier) && (fscore <= alpha)
    & (opposing_pieces(current) > 3)) depth = pre_frontier;
    /* decide about extended futile pruning at pre-frontier nodes */
    fscore = (mat_balance(current) + extd_futil_margin);
    if (!extend && (depth == pre_frontier) && (fscore <= alpha))
    { fprune = 1, score = fmax = fscore, }
    /* decide about selective futile pruning at frontier nodes */
    fscore = (mat_balance(current) + futil_margin);
    if (!check(move) && (depth == frontier) && (fscore <= alpha))
    { fprune = 1, score = fmax = fscore, }
    /* probe the transposition tables at the current node */
    tt_hit = probe_transposition_tables(current, depth, tt_ref);
    if (tt_hit) { ... } else { ... }
    /* try the adaptive null-move search with a minimal window around */
    /* "beta" only if it is allowed, desired, and really promises to cut off */
    if (!fprune && !check(move) && null_okay(current, move)
    && try_null(alpha, beta, current, depth, move, tt_ref)) {
        null_score = -search(-beta, -beta + 1, null_move, current,
        depth - R_adpt(current, depth) - 1);
        if (null_score >= beta) return null_score;
    } /* fail-high null-move cutoff */
}
/* now continue as usual but prune all futile moves if "fprune = 1" */
for (move = first(current), (move != 0), move = next(current, move))
    if (!fprune || check(move) /* recursive PV search */
    || (fmax + mat_gain(move) > alpha)) { ... }
    else fpruned_moves++;
/* "fpruned_moves != 0" => the search was selective at the current node */
}

Figure 1: Selective Search with Combined AEL Pruning.
AEL Pruning

<table>
<thead>
<tr>
<th>Test Suite</th>
<th>Normal #Nodes</th>
<th>Normal #Solved</th>
<th>+AEL Δ Nodes</th>
<th>+AEL Δ Solved</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECM-08</td>
<td>1,232,004,798</td>
<td>552 / 879</td>
<td>-18.82%</td>
<td>-5</td>
</tr>
<tr>
<td>ECM-10</td>
<td>8,821,781,692</td>
<td>642 / 879</td>
<td>-29.66%</td>
<td>-4</td>
</tr>
<tr>
<td>ECM-12</td>
<td>83,443,531,950</td>
<td>704 / 879</td>
<td>-49.48%</td>
<td>-6</td>
</tr>
<tr>
<td>WAC-08</td>
<td>146,094,041</td>
<td>285 / 300</td>
<td>-24.06%</td>
<td>+1</td>
</tr>
<tr>
<td>WAC-10</td>
<td>946,867,509</td>
<td>296 / 300</td>
<td>-39.93%</td>
<td>-1</td>
</tr>
<tr>
<td>WAC-12</td>
<td>8,998,551,515</td>
<td>296 / 300</td>
<td>-57.69%</td>
<td>0</td>
</tr>
<tr>
<td>WCS-08</td>
<td>750,804,397</td>
<td>841 / 1001</td>
<td>-23.05%</td>
<td>-1</td>
</tr>
<tr>
<td>WCS-10</td>
<td>5,398,696,585</td>
<td>866 / 1001</td>
<td>-34.91%</td>
<td>-2</td>
</tr>
<tr>
<td>WCS-12</td>
<td>52,801,555,626</td>
<td>874 / 1001</td>
<td>-49.38%</td>
<td>-4</td>
</tr>
<tr>
<td>Sum-08</td>
<td>2,128,903,236</td>
<td>1678 / 2180</td>
<td>-20.67%</td>
<td>-5</td>
</tr>
<tr>
<td>Sum-10</td>
<td>15,169,345,786</td>
<td>1804 / 2180</td>
<td>-32.17%</td>
<td>-7</td>
</tr>
<tr>
<td>Sum-12</td>
<td>145,243,639,091</td>
<td>1874 / 2180</td>
<td>-49.95%</td>
<td>-2</td>
</tr>
</tbody>
</table>

Table 1. Performance of Normal vs. AEL-Pruning DARK THOUGHT.

<table>
<thead>
<tr>
<th>Test Suite</th>
<th>Time</th>
<th>Result</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>BS-2830 (test rating = 2734 points)</td>
<td>15 min.</td>
<td>18 / 27</td>
<td>391K apps</td>
</tr>
<tr>
<td>BT-2830 (test rating = 2526 points)</td>
<td>15 min.</td>
<td>27 / 30</td>
<td>440K apps</td>
</tr>
<tr>
<td>LCT-H (test rating = 2645 points)</td>
<td>10 min.</td>
<td>28 / 35</td>
<td>401K apps</td>
</tr>
<tr>
<td>Bratko / Kpec</td>
<td>1 min.</td>
<td>18 / 24</td>
<td>362K apps</td>
</tr>
<tr>
<td>Encyclopedia of Chess Middlegames</td>
<td>5 sec.</td>
<td>534 / 879</td>
<td>401K apps</td>
</tr>
<tr>
<td>Win at Chess</td>
<td>5 sec.</td>
<td>296 / 300</td>
<td>446K apps</td>
</tr>
<tr>
<td>World's Greatest Chess Games</td>
<td>5 sec.</td>
<td>23 / 50</td>
<td>368K apps</td>
</tr>
<tr>
<td>1001 Winning Chess Sacrifices</td>
<td>5 sec.</td>
<td>832 / 1001</td>
<td>435K apps</td>
</tr>
</tbody>
</table>

Table 2. Test-Suite Results of DARK THOUGHT with AEL Pruning.

4. TEST GAMES

We played 580 test games with DARK THOUGHT in order to assess the real playing capabilities of AEL pruning as resulting from both its tactical and positional skills. 300 of the games featured self-play at different time controls between the normal DARK THOUGHT and the otherwise identical version of DARK THOUGHT with AEL pruning (see Section 4.1). The remaining 280 games pitted DARK THOUGHT with AEL pruning against 20 strong PC chess programs at tournament-style time controls (see Section 4.2). We chose balanced starting positions from the Encyclopedia of Chess Openings (ECO) for all games and played two games from every single starting position with swapped colours of the according opponents.

4.1 Self-Play

In general, forward-pruning schemes like AEL pruning suffer most from unsound cutoffs at shallow overall search depth. Hence, we ran 100 self-play games each at fixed iteration depths of 8 and 10 plies. In contrast thereto, the last 100 self-play games measured the performance of AEL pruning in standard play by enforcing tournament-style time controls at a rate of 60 moves in 90 minutes for both equally equipped opponents. Table 3 summarizes the final results and winning percentages of all self-play games. Appendix E gives the complete move lists of 10 representative self-play games for each kind of time control as played from the so-called “Nuan Positions” 42 (ECO B89), 43 (ECO C19), 45 (ECO D36), 47 (ECO B15), and 49 (ECO A25).
Table 3: DARK THOUGHT with AEL Pruning vs Normal DARK THOUGHT.

Although 100 games do not suffice to quantify the difference in playing strength between two opponents with good statistical confidence, the self-play results strongly suggest that AEL pruning does not measurably weaken DARK THOUGHT at fixed search depths. Even for rather shallow fixed depths, the addition of AEL pruning to DARK THOUGHT did not result in any notable decrease of playing strength. While DARK THOUGHT with AEL pruning visited far fewer nodes than the normal version, it achieved almost perfectly even scoring percentages of 49.0% and 50.5% for self-play at fixed depths of 8 and 10 plies respectively (see Table 3). The strong self-play performance of AEL pruning at fixed iteration depths is consistent with the overall results of our test-suite experiments from Section 3.2.

In standard play at tournament-style time controls, the smaller search trees of AEL pruning pay off handily as evident from the 67.5% winning percentage listed in Table 3. This high overall score shows DARK THOUGHT with AEL pruning as the truly better player than the normal version with 95% statistical confidence. On a standard chess-rating scale based on relative performances (Mysliwetz, 1994), the winning percentage of 67.5% indicates that the visibly stronger play of DARK THOUGHT with AEL pruning might be worth about 100 rating points in self-play at tournament time-controls versus the normal DARK THOUGHT. Self-play experiments by other researchers quantified such a rating increase to be equivalent to roughly twice the search speed (Szabo and Szabo, 1988, Mysliwetz, 1994). Taken the other way round, the 67.5% self-play winning percentage has at an average reduction of search effort by half for DARK THOUGHT with AEL pruning. This is consistent with the 12-ply results of our test-suite experiments from Section 3.2. And indeed, DARK THOUGHT with AEL pruning routinely reaches iteration depths of 11 to 13 plies in the middlegame at tournament play.

Table 4: Strong PC Chess Programs vs. AEL Pruning DARK THOUGHT.

<table>
<thead>
<tr>
<th></th>
<th>DT #2</th>
<th>DT #3</th>
<th>DT #4</th>
<th>DT #5</th>
<th>DT #7</th>
<th>DT #8</th>
<th>DT #9</th>
<th>Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>10:13</td>
</tr>
<tr>
<td>CM</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>=</td>
<td>+</td>
<td>5:0:90</td>
</tr>
<tr>
<td>CR</td>
<td>=</td>
<td>-</td>
<td>+</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>-</td>
<td>5:0:90</td>
</tr>
<tr>
<td>F3</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>2:0:12</td>
</tr>
<tr>
<td>H5</td>
<td>=</td>
<td>-</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>5:5:85</td>
</tr>
<tr>
<td>FB</td>
<td>=</td>
<td>=</td>
<td>-</td>
<td>+</td>
<td>=</td>
<td>=</td>
<td>-</td>
<td>3:5:10</td>
</tr>
<tr>
<td>G3</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>-</td>
<td>=</td>
<td>+</td>
<td>=</td>
<td>5:0:90</td>
</tr>
<tr>
<td>G5</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>2:5:15</td>
</tr>
<tr>
<td>H6</td>
<td>=</td>
<td>=</td>
<td>+</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>4:5:95</td>
</tr>
<tr>
<td>J4</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>3:0:11</td>
</tr>
<tr>
<td>J5</td>
<td>=</td>
<td>=</td>
<td>+</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>4:5:95</td>
</tr>
<tr>
<td>M5</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>3:0:11</td>
</tr>
<tr>
<td>M7</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>+</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>4:5:95</td>
</tr>
<tr>
<td>MM</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>1:5:12</td>
</tr>
<tr>
<td>M8</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>1:5:12</td>
</tr>
<tr>
<td>N8</td>
<td>+</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>4:0:10</td>
</tr>
<tr>
<td>N9</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>4:5:95</td>
</tr>
<tr>
<td>R8</td>
<td>+</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>4:0:10</td>
</tr>
<tr>
<td>R9</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>2:5:11</td>
</tr>
<tr>
<td>S8</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>3:5:10</td>
</tr>
<tr>
<td>Sum</td>
<td>10:5:29.3</td>
<td>11:5:28.5</td>
<td>10:0:30.0</td>
<td>10:5:29.3</td>
<td>9:0:31.0</td>
<td>8:5:31.5</td>
<td>10:5:29.5</td>
<td>70:5:208.5</td>
</tr>
</tbody>
</table>
4.2 Nunn Matches

We played a series of 280 test games of DARKTHOUGHT with AEL pruning versus 20 other strong chess programs, among them the latest releases of current and former world champions (e.g., Fritz, Genius, and Junior). All games were played with DARKTHOUGHT consuming roughly 170 MBytes of memory and running on a 600 MHz DEC Alpha-21164a LX164 workstation. The opponents enjoyed the service of our then fastest PC, hosting a 233 MHz AMD K6 CPU and 64 MBytes of RAM. We list the names and shorthand notations of the participating PC programs in alphabetical order below (see Appendix D of Heinz (2000) for more detailed information about the exact configurations).

Chessmaster 4000 Turbo (CM), Comet A 86/ A 90 (CO), Crafty 16.4 (CR), Fritz 3.10 (F9), Fritz 5.0/5.03 (F5), Fritz 5.32b (F3), Genius 5 (G5), Genius 5 (G5), Harcs 60 (H6), Junior 4.6 (J4), Junior 5.0 (J5), MChess 5 (M5), MChess 7.1 (M7), Mephisto (MM), Mephisto Shredder (M), Nimzo 98 (N8), Nimzo 99a (N9), Rebel 8 (R8), Rebel 9 (R9), Shredder 2 (S2).

We started the 280 test games from the so-called "Nunn Positions" #2 (ECO B89), #3 (ECO C19), #4 (ECO C97), #5 (ECO D36), #7 (ECO E15), #8 (ECO E98) and #9 (ECO A25) with all opening books and learning features of the programs disabled. In order to partly account for the hardware advantage of the 600 MHz DEC Alpha-21164a (18.4 SPECint95) over the 233 MHz AMD K6 (9.6 SPECint95), DARKTHOUGHT played at a rate of 60 moves in 90 minutes while the PC programs received additional 30 minutes on their clocks (60 moves in 120 minutes). We decided against awarding exactly double the time to the PC programs because DARKTHOUGHT indirectly benefits from longer search times of the opponents whenever it correctly predicts and poisons their moves. Our experience with hundreds of test games suggests that a time increase by a factor of 1.5 strikes a good balance in this respect. The test games ended 209.5-65.5 in favour of DARKTHOUGHT (see Table 4). The overall winning percentage of 74.82% provides further qualitative empirical evidence for the sustained tactical abilities and the practical usefulness of AEL pruning in standard play. Appendix D of Heinz (2000) contains the complete move lists of all the 280 Nunn test games featuring DARKTHOUGHT versus other strong chess programs.

5. CONCLUSION

This article introduced AEL pruning which combines the three techniques of adaptive null-move pruning, extended futility pruning, and limited reasoning. Our thorough investigations of the practical behaviour and effectiveness of AEL pruning revealed that it really enables "forward pruning without tears" in computer chess. The overall results of our test games and test-suite experiments hint at an increase in playing strength of roughly 100 rating points for the addition of AEL pruning to our tournament proven chess program DARKTHOUGHT. Nevertheless, combined AEL pruning is easy and efficient to implement as demonstrated by our detailed algorithmic description of the new technique within about 50 lines of ANSI-C code.

Our test results undoubtedly add AEL pruning to the collection of practically successful forward-pruning techniques in computer chess. In particular, the extensive experiments with 2180 test positions from well-known test suites showed that AEL pruning preserves the tactical strength of DARKTHOUGHT while reducing its search effort by 20 to 50 percent on average at fixed iteration depths of 8 to 12 plies. The reduction in search effort scales equally well with progressing search depth as that for any of the three constituent techniques alone. The positive results of our 580 test games (self-play and versus other strong chess programs) provide further empirical evidence for the practical usefulness of AEL pruning in real game-play.
6. REFERENCES


7. APPENDICES

APPENDIX A: EXPERIMENTAL SETUP

- Test suites *Encyclopedia of Chess Middlegames* (ECM, 879 positions), *Win at Chess* (WAC, 300 positions), and 1001 *Winning Chess Sacrifices* (WCS, 1001 positions) as available on the Internet.
- **Dark Thought** as of March 31, 1998 with 8M transposition-table entries (4M per side), 1M King hash-table entries (512K per side), and 512K Pawns hash-table entries.
- Digital Unix 4.0d program development tools and operating system.
- 600 MHz DEC Alpha-21164a LX164 workstation (600 MHz DEC Alpha-21164a CPU, 8KB/8KB on-chip L1/L2 cache, 96KB unified on-chip L2 cache, 2MB unified off-chip L3 cache, DEC LX164 mainboard, 2x128MB SDRAM DIMMs = 256MB RAM).

APPENDIX B: SELECTED SELF-PLAY GAMES

10 Games Played at 60 Moves in 90 Minutes (Result: AEL Pruning - Normal = 6.5:3.5 Points)

**AEL Pruning - Normal:** [L1-L2] (Self-Play Test Game, Normal Position #2, ECO B89, June 1998)
1 e 4 e 5 Nf 3 Nc 6 d 4 exd 4 Nxd 4 Nc 6 d 6 Nf 6 Bc 4 7 Bc 4 e 5 Qc 2 Qc 5 Qf 5 g 5 Qg 5 h 4 Qh 5 g 5 e 5 f 4 exd 5 20 h 6 Qb 24 Qb 5 Qb 24 Qg 5 Rd 25 b 3 Kbd 26 Qb 5 27 Re 5 Ne 5 26 g 5 Qe 6 20 exd 5 exd 5 21 Kd 5 21 e 6 g 6 e 6 20 Qd 6 Qbd 24 Nbd 24 Nbd 25 Nbd 26 Nbd 5 27 Kd 1 28 Re 3 30 Kf 5 31 1/2 1/2

**AEL Pruning - Normal:** [L1-L2] (Self-Play Test Game, Normal Position #3, ECO B89, June 1998)
1 e 4 e 5 Nf 3 d 4 e 5 Nxd 5 Nf 5 d 4 Nf 6 Bc 5 6 Bc 5 6 Qe 4 6 Qe 4 5 c 6 c 7 7 Qc 7 7 Nf 5 Ne 7 8 a 4 b 5 b 6 Bd 7 10 Bd 7 Nbd 11 10 c 6 c 6 exd 6 5 13 Nc 14 Bc 4 c 5 Qd 8 10 a 5 b 5 17 Bc 5 Qd 3 13 exd 3 a 9 Be 7 Ke 7 20 Rac 1 Kc 1 21 Kc 1 Be 7 22 Re 8 Be 8 23 Kf 4 b 4 24 Kf 4 b 5 25 Kf 4 b 5 26 e 6 a 6 27 Nc 7 28 e 6 29 e 7 30 Nf 5 a 5 31 Nf 5 d 6 20 f 5 41 e 2 42 Nc 6 53 54 Kc 2 55 Nc 2 56 Kc 2 57 Kc 2 58 Nc 5 59 Kd 6 60 Kc 5 61 Kc 5 62 Kd 8 63 Kc 5 64 Kc 5 65 Ke 7 66 Ke 7 68 Ke 7 68 Ke 7 67 Ke 7 68 Ke 7 69 Ke 7 70 Ke 8 7 71 Ke 8 7 72 Ke 8 7 73 Ke 8 7 74 Ke 8 7 75 Ke 8 7 76 Ke 8 7 77 Ke 8 7 78 Ke 8 7 79 Kg 8 80 Kg 8 81 Kg 8 82 Kg 8 1/2 1/2

**AEL Pruning - Normal:** [L1-L2] (Self-Play Test Game, Normal Position #3, ECO C19, June 1998)
1 e 4 e 5 Nf 3 d 4 c 5 6 Bb 5 Bc 4 6 Bc 4 5 c 6 c 7 7 Qc 7 7 Nf 5 Ne 7 8 a 4 a 4 9 b 5 Bd 7 10 Bd 7 Nbd 11 10 c 6 c 6 exd 6 5 13 Nc 14 Bc 4 c 5 Qd 8 10 a 5 b 5 17 Bc 5 Qd 3 13 exd 3 a 9 Be 7 Ke 7 20 Rac 1 Kc 1 21 Kc 1 Be 7 22 Re 8 Be 8 23 Kf 4 b 4 24 Kf 4 b 5 25 Kf 4 b 5 26 e 6 a 6 27 Nc 7 28 e 6 29 e 7 30 Nf 5 a 5 31 Nf 5 d 6 20 f 5 41 e 2 42 Nc 6 53 54 Kc 2 55 Nc 2 56 Kc 2 57 Kc 2 58 Nc 5 59 Kd 6 60 Kc 5 61 Kc 5 62 Kd 8 63 Kc 5 64 Kc 5 65 Ke 7 66 Ke 7 68 Ke 7 68 Ke 7 67 Ke 7 68 Ke 7 69 Ke 7 70 Ke 8 7 71 Ke 8 7 72 Ke 8 7 73 Ke 8 7 74 Ke 8 7 75 Ke 8 7 76 Ke 8 7 77 Ke 8 7 78 Ke 8 7 79 Kg 8 80 Kg 8 81 Kg 8 1/2 1/2

**AEL Pruning - Normal:** [L1-L2] (Self-Play Test Game, Normal Position #5, ECO D36, June 1998)
1 d 4 e 5 2 6 4 3 e 5 f 3 e 6 e 4 5 exd 5 5 Ne 5 6 Bg 5 Bf 7 7 Qc 2 Qb 6 7 8 e 5 9 Ne 9 Bd 3 Re 8 10 0 f 6 f 6 10 11 12 Bb 1 12 13 Na 14 Ne 15 Bc 7 16 Ne 5 f 5 16 a 4 Kf 8 17 Kf 1 18 19 20 21 20 21 22 23 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 1 2 1 2

**AEL Pruning - Normal:** [L1-L2] (Self-Play Test Game, Normal Position #5, ECO D36, June 1998)
1 d 4 e 5 2 6 4 3 e 5 f 3 e 6 e 4 5 exd 5 5 Ne 5 6 Bg 5 Bf 7 7 Qc 2 Qb 6 7 8 e 5 9 Ne 9 Bd 3 Re 8 10 0 f 6 f 6 10 11 12 Bb 1 12 13 Na 14 Ne 15 Bc 7 16 Ne 5 f 5 16 a 4 Kf 8 17 Kf 1 18 19 20 21 20 21 22 23 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 1 2 1 2

No license. PDF produced by Phyllis R. Egan - http://www.thes-net.com/elf/paragraphs
10 Games Played at a Fixed Depth of 10 Piles (Result: AEL Pruning - Normal = 5.0 : 5.0 Points)

AEL Pruning - Normal [F - L] (Self-Play Test Game, July 1999)
