A New Self-Play Experiment in Computer Chess

[M.I.T. LCS Technical Memo 608 / MIT-LCS-TM-608]

Ernst A. Heinz

M.I.T. Laboratory for Computer Science (Room NE 43 – 228)
Massachussetts Institute of Technology
545 Technology Square, Cambridge, MA 02139, USA

Email = <heinz@mit.edu>, WWW = <http://supertech.lcs.mit.edu/~heinz/>

May 20, 2000

Content Areas: computer chess, self-play experiments

Abstract. This paper presents the results of a new self-play experiment in computer chess. It is the first such experiment ever to feature search depths beyond 9 plies and thousands of games for every single match. Overall, we executed 17,150 self-play games (1,050–3,000 per match) in one "calibration" match and seven "depth X+1 ⇔ X" handicap matches at fixed iteration depths ranging from 5–12 plies. For the experiment to be realistic and independently repeatable, we relied on a state-of-the-art commercial contestant: Fritz 6, one of the strongest modern chess programs available. The main result of our new experiment is that it shows the existence of diminishing returns for additional search in computer chess self-play by Fritz 6 with 95% statistical confidence. The diminishing returns manifest themselves by declining rates of won games and reversely increasing rates of drawn games for the deeper searching program versions. The rate of lost games, however, remains quite steady for the whole depth range of 5–12 plies.

1 Introduction

To the best of our knowledge, Gillogly and Newborn in 1978 independently reported the earliest attempts at modeling the relationship between the playing strength of chess programs on one hand and the available computing power or search depth on the other. Gillogly [4] introduced his "technology curve" that plotted the playing strength against what he called "machine power" on a logarithmic scale. Newborn [21, 22] related the numbers of nodes as searched by different chess programs in three minutes (the average time per move in tournament games) to the playing strengths of the very same programs as derived from their actual performances in tournaments. Later on, Levy [17] and Levy and Newborn [16] refined Newborn's initial scheme by contrasting the highest rated tournament performances of the best chess programs with the years of their achievement. All these comparisons inevitably led to speculative extrapolations

Table 1. Timeline of published self-play experiments in computer chess. [TECHMATE and ZUGZWANG self-played with time handicaps.]

Year	Program	Experimenter	Depths (in Plies)	Δ ELO (per Ply)	No. Games	
					All	Each
1982	BELLE	Thompson	3 – 8	+246	100	20
1983	BELLE	Condon, Thompson	4 – 9	+217	300	20
1988	TECHMATE	Szabo, Szabo	– –	– –	6,882	$\geq$ 32
1990	HITECH LOTECH	Berliner et al.	4 – 9	+195 +232	1,056	16
1994	ZUGZWANG	Mysliwietz	– –	– –	450	50
1996	PHOENIX	Schaeffer	4 – 9	+228	120	20
1997	THE TURK	Junghanns et al.	3 – 9	+200	480	80

which Levy characterized as the "meta-science of prediction in computer chess" in his latest article [15] about the subject in 1997.

In the early 1980s, Thompson [3, 28] pioneered the usage of self-play with his then reigning World Computer-Chess Champion machine BELLE. Self-play with handicaps in search depth, search speed, or search time between otherwise identical program versions represents a more rigorous approach of investigating the relationship of computing power and the strength of chess programs. A notable advantage of such matches is that the scoring rates quantify the differences in playing strength of the various participating versions of the same program. Despite unresolved questions regarding the magnitude of self-play rating differences [1], self-play seems to be the best of the available methods to resolve the old but still ongoing "search versus knowledge" debate [11, 24, 25]. Almost everybody seems to agree with the intuitive notion that the positive effect of more search ought to taper off with increasing overall search effort. In self-play matches such "diminishing returns for additional search" should lead to significantly lower scoring rates of the deeper searching program versions with the progression towards higher search depths.

However, Thompson's experiments [3, 28] led to the surprising result that the playing strength of BELLE increased almost linearly with search depth. For fixed-depth searches of 3–9 plies, the increase in playing strength amounted to roughly 200 ELO rating points per ply. Several other researchers later confirmed Thompson's findings by self-play experiments with their own chess programs HITECH, LOTECH, PHOENIX, and THE TURK [1, 12]. In Figure 1 of their article [12], Junghanns et al. showed that the scoring rates of the program versions searching one ply deeper remained range-bound between 70%–80% in all cases. There are no clearly visible average downward trends at the end of these 9-ply data curves.

1.1 Previous Self-Play Experiments in Computer Chess

Table 1 presents an overview and timeline of self-play experiments in computer chess published up to now. Beside names, depths, and average ELO increases, the table also lists the overall numbers of games played in the experiments as a whole and for each single match. Unfortunately, all the experiments feature only very low numbers of games per match which do not allow for any confident quantification of rating differences between the opponents. Hence, we completely agree with Mysliwietz [19] who criticized the statistical uncertainty of self-play experiments in computer chess already back in 1994.

Based on this criticism, we re-assessed and carefully re-analyzed all experiments from Table 1 in our recent publications [6, 8]. The outcome of our analyses was bleak and simple: none of the previous self-play experiments provide any confident quantifications of the differences in playing strength. The experimental results are just not statistically significant. The experiments do not feature enough games per match to draw reliable conclusions. Based on rigorous analyses of hypothetical match results, we conjectured [6, 8] that at least 1,000 games per match are necessary to assess diminishing returns in computer self-play with 95% statistical confidence. Further questions regarding previous self-play experiments are the exact meaning of "fixed depth" in each case, the details of the experimental setups, and their repeatability.

The "fixed depth" question is not trivial because the modes of operation of the programs differ substantially depending on its real meaning. In the search-theoretical sense, "fixed depth" denotes true brute-force search with uniform path lengths from the root to all horizon nodes and no selectivity at all – neither by means of depth reductions or other kinds of forward pruning nor by any search extensions. In computer-chess practice, however, "fixed depth" usually equals "fixed iteration depth" which relates to the depth limit of iterative deepening [13, 26] as performed by the top-level search control. Here, the programs operate with an iteration limit instead of a time bound but otherwise execute their sophisticated variable-depth search procedure as built in – with all kinds of depth reductions, forward pruning, and search extensions fully enabled.

The exact setups of the experiments are not only important for the purpose of repeatability. The engine settings and hash-table sizes, the opening books or positions used, the endgame databases, etc. may as well have non-negligible influences on the match results. Of course, the whole setup must be identical for all sibling versions of a program during handicap self-play. In particular, it is not admissible to increase the hash-table sizes of deeper searching versions in order to speed-up their times to completion.

As for repeatability, Table 1 shows that all self-play experiments published up to now featured proprietary chess program. Several of them also relied on special hardware (Belle, Hitech, Lotech, and Zugzwang). Hence, even assuming detailed knowledge of the exact setups, none of the experiments was independently repeatable by others in practice.

1.2 Our New Self-Play Experiment in Computer Chess

We designed our new self-play experiment in such a way as to overcome the aforementioned drawbacks of its predecessors from Table 1. Our primary concerns were the rigorous analysis of the results (see Section 3) and their statistical significance. We played seven "depth X+1 ⇔ X" handicap matches at fixed iteration depths ranging from 5–12 plies with 1,050–3,000 games per match (see Section 5). By extending the self-play depths beyond 9 plies for the first time ever, we sought to gain new information about potentially diminishing returns for additional search in computer chess at high depths.

Moreover, we intended our self-play experiment to be transparent and realistic at the time of execution and independently repeatable by others later on. To this end, we needed a state-of-the-art contestant featuring general worldwide availability, x86-PC compatibility, well-defined parameter control, and – last but not least – handicap self-play ability. The commercial chess program FRITZ 6 (written by Frans Morsch and Matthias Feist) met all our requirements because handicap self-play abilities were included in it upon our special request. FRITZ 6 is certainly one of the strongest modern chess programs available.

Further advantages of employing FRITZ 6 spring from its database capabilities, versatile chess-engine concept, and excellent opening book (composed by Alexander Kure). In particular, the wide and well-balanced opening book facilitates the automatic play of fair matches with thousands of games. Just to be sure, we checked the integrity and Black / White fairness of the opening book by means of a "calibration" match between two identical opponents (see Section 4.1). The various different engines available for the FRITZ 6 interface allow for the possibility to include other chess programs in our future self-play research.

2 Related Work

In the introduction, we already mentioned the attempts of Gillogly [4], Newborn [21, 22], Levy [15, 17], and Levy and Newborn [16] at modeling the relationship between playing strength and computing power. Newborn [20] introduced yet another technique to study this relationship in 1985. The rationale of Newborn's novel approach sprang from the assumption that new best moves as discovered by chess programs at higher search depths ought to represent better choices than the best moves preferred at shallower depths. To this end, Newborn tracked the behaviour of BELLE for searches to fixed depths of 11 plies on a set of 447 test positions from real games. Interestingly enough, his data correlated closely with Thompson's earlier self-play results of BELLE [3, 28].

In 1997, Junghanns et al. [12] let PHOENIX and THE TURK search roughly 1,000 positions from self-play games to fixed depths of 9 plies while recording new best moves beside other information. Also during 1997, Hyatt and Newborn [10] conducted another behavioural experiment with Hyatt's chess program CRAFTY searching 347 new test positions to fixed depths of 14 plies. This experiment revealed the astonishing fact that the rate of new best moves as chosen by CRAFTY at high search depths of 9–14 plies remained quite steady around 15%–17% on average and hardly decreased anymore. Following up thereon, we confirmed Hyatt and Newborn's findings by repeating their "go deep" experiment with our own chess program DARKTHOUGHT in 1998 [6, 9]. Recently, we pushed the limits of going deep to fixed depths of 16 plies [6, 7] where the best-change rate of DarkThought still remains steady at roughly 15%.

Self-play with handicaps in search depth, search speed, or search time between otherwise identical program versions is a valuable tool not only for computer chess but for computer strategy game-playing in general. Examples from other domains than chess include self-play experiments in computer checkers by Schaeffer et al. [11, 12, 23, 24] with the reigning World Man-Machine Checkers Champion Chinook as well as self-play experiments in computer Othello by Lee and Mahajan [14] with their program Bill and by Brockington et al. [2, 11, 12] with his Othello program Keyano. For all published self-play experiments in computer chess (see also Table 1) we provide further descriptions below.

1982: Thompson [28]. Thompson's pioneering experiment featured 100 self-play games with matches of 20 games each between versions of Belle differing by exactly one ply in lookahead for fixed depths of 3–8 plies. The gain in playing strength averaged at 246 rating points per ply of search. The experiment showed no diminishing returns at any depth.

1983: Condon and Thompson [3]. In the second experiment, Condon and Thompson let Belle self-play 300 games in round-robin style with matches of 20 games each between all program versions for fixed depths of 4–9 plies. The gain in playing strength averaged at 217 rating points per ply of search. The observed ratings slightly hinted at limited diminishing returns from a fixed depth of 6 plies onwards. Yet, the results of the experiment are not statistically significant.

1988: Szabo and Szabo [27]. The Szabos determined the technology curve of their chess program TechMate that self-played 6,882 games on two Atari ST computers. The number of games per match between longer and shorter searching versions of the program varied strongly from a minimum of 32 to a maximum of 1367. The gain in playing strength averaged at 156 rating points per doubling of available search time (computing power). The experimental data indicated slight diminishing returns at longer search times. However, the Szabos simply did not play enough games at long times to draw reliable conclusions.

1990: Berliner et al. [1]. The Hitech team made their chess machine self-play 1,056 games in a round-robin setting with matches of 16 games each between all program versions of Hitech and Lotech (a variant of Hitech scaled down knowledge-wise) for fixed depths of 4–9 plies. The gain in playing strength averaged 195 rating points per ply of search for Hitech and 232 rating points per ply for Lotech. The ratings showed possible signs of limited diminishing returns starting at a fixed depth of 6 plies. But there was no clear trend of diminishing returns at higher search depths and the experimental results are not statistically significant.

1994: Mysliwietz [19]. Mysliwietz let the parallel chess program Zugzwang self-play 450 games with 50 games per match between program versions that differed roughly by a factor of two in search speed due to varying numbers of allotted processors. The gain in playing strength averaged 109 rating points per doubling of search speed for 9 successive doubling steps. The observed ratings do not exhibit any diminishing returns at all.

1996: Schaeffer [12]. Junghanns et al. [11, 12] briefly mentioned the results of a self-play experiment by Schaeffer with his chess program PHOENIX in 1996. The experiment comprised 120 self-play games with matches of 20 games each between program versions that differed by exactly one ply in lookahead for fixed depths of 3–9 plies. The gain in playing strength averaged at 228 rating points per ply of search. The result of the "9 $\Leftrightarrow$ 8" match might be interpreted as an indication of diminishing returns. Yet, a single error-prone data point like this at the end of the curve really lacks significance.

1997: Junghanns et al. [12]. The self-play experiment with Björnsson and Junghanns' chess program THE TURK featured 480 games with matches of 80 games each between program versions differing by exactly one ply in lookahead for fixed depths of 3–9 plies. The gain in playing strength averaged around 200 rating points per ply of search. The scoring rates of the deeper searching versions of THE TURK actually increased steadily from fixed search depths of 6 plies onwards, thus even hinting at additional gains in returns for higher search depths rather than diminishing ones.

Junghanns et al. continued to look for diminishing returns by means of other metrics than self-play in [12]. They finally claimed to have found empirical evidence in this respect. According to their explanations, the low search quality of chess programs (i.e. their high error probability) and the abnormally large lengths of self-play games inadvertently hide diminishing returns in computer chess (which doubtlessly exist in their opinion). Although we greatly appreciate Junghanns et al.'s trial aimed at the better understanding of diminishing returns in computer chess, we are not convinced that their claims hold when subjected to rigorous methodological and statistical testing. Hence, the quest for indisputable and statistically significant demonstrations of diminishing returns for additional search in computer chess still remained to be concluded.

3 Statistical Analysis of Self-Play Experiments

In our recent publications [6, 8] we introduced a general mathematical framework for the statistical confidence analysis of self-play experiments. Based on this framework, we scrutinized the self-play data published by other researchers for computer chess, computer checkers, and computer Othello (see Chapter 9 of [6]). Of course, we apply the same framework here to analyze our own new self-play results. For the sake of completeness, we explain the underlying fundamentals and notations of the framework in brief below.

We call $w = x/n$ the *scoring rate* which results from a score of $x \leq n$ points in a match or tournament of n games. The scoring rate $0 \leq w \leq 1$ estimates a player's real *winning probability* in games versus the respective opponents. Therefore, we may simply assume the scoring rate to be the sample mean of a binary-valued random variable that counts two draws as a loss plus a win. This enables the calculation of standard errors and %-level confident bounds for any match results by applying classical statistics [5, 18] to the values of x and n.

Standard Errors of Scoring Rates. The standard error $s(w)$ of a scoring rate $w = x/n$ is given by $s(w) = \sqrt{w * (1-w)/n}$.

Confident Bounds on Winning Probabilities. Let $z_{\%}$ denote the upper critical value of the standard $N(0,1)$ normal distribution for any desired %-level of statistical confidence ($z_{90\%} = 1.645$, $z_{95\%} = 1.96$).

- $w \pm z_{\%} * s(w)$

places %-level confident lower and upper bounds on the real winning probability of a player with scoring rate $w = x/n$. [Remark: The bounds are accurate only if $x > 4$ and $n - x > 4$. Otherwise, the sample data does not provide enough information for the determination of statistically confident bounds. In such cases the approximate bound values as calculated by the given formula underestimate the real deviations possible.]

Confident Bounds on Differences of Winning Probabilities. From the above we derive %-level confident lower and upper bounds on the difference in real winning probability between two players with scoring rates $w_1 = x_1/n_1$ and $w_2 = x_2/n_2$ where $w_1 \geq w_2$.

- $l_{\%} = \max\Big((w_1 - z_{\%} * s(w_1)) - (w_2 + z_{\%} * s(w_2)),\ \ -1\Big)$
- $u_{\%} = \min\Big((w_1 + z_{\%} * s(w_1)) - (w_2 - z_{\%} * s(w_2)),\ \ +1\Big)$

For these bounds it holds that $-1 \leq l_{\%} \leq u_{\%} \leq 1$ and $u_{\%} \geq 0$. We denote the range $[l_{\%}, u_{\%}]$ by *%-level confident Δw*. The tables of this paper also refer thereto by "90%-C Δw" and "95%-C Δw" in their column heads.

The bounds allow for confident quantifications of differences in playing strength between two players as measured by their winning probabilities. Whenever $l_{\%} > 0$ we are %-level confident that the player with the higher scoring rate is indeed stronger than the other. If $l_{\%} \leq 0$, however, we cannot discriminate the two players' strengths with the desired confidence: the supposedly weaker player with the lower scoring rate might really be as strong as the other or even stronger.

Our self-play matches test the playing strengths of successive program versions on the scale of increasing search depths. We use the scoring rates w_1 of the match winners and $w_2 = 1 - w_1$ of the losers for our calculations of $l_{\%}$ and $u_{\%}$. After determining w and $s(w)$, we calculate the %-level confident ranges $[l_{\%}, u_{\%}]$ for all consecutive matches and call their intersection $[\Delta w]_{\%}$. If $[\Delta w]_{\%} = \emptyset$ (empty intersection) we are %-level confident that the differences in real winning probability and, thus, playing strength of successive program versions cannot be identical for all tested ones. Then, the overall results refute the notion of constant returns for additional search throughout the whole experiment with the

Table 2. Detailed engine setup of FRITZ 6 for the self-play matches.

Book Choice	"General.ctg"
Book Options	use tournament book = on, use book = on minimum games = 2 variety of play = maximal (++) influence of learn value = none (−−) learning strength = none (−−)
Use Tablebases	off
Engine Parameters	contempt value = 0, aggressiveness = 0 selectivity = 2, tablebase depth = 0
Hashtable Size	32 MB

desired %-level of confidence. Otherwise, the union $[L_{\%}, U_{\%}] = \bigcup [l_{\%}, u_{\%}]$ of all confident bound ranges confirms constant or at least nearly constant returns for additional search of the tested program if $U_{\%} - L_{\%} < \epsilon$ for some small $\epsilon \geq 0$.

4 Experimental Setup

In our self-play matches the initially released version of the FRITZ 6 engine dated November 10, 1999 (size: 291,328 bytes) competed against itself. All opponents relied on the opening book "General.ctg" from the original FRITZ 6 CD-ROM with tournament mode and maximal variety of play activated. We disabled the book learning, tablebase access (no endgame databases installed), permanent brain, and early resign options. Moreover, we set the contempt values of the engines to zero. The detailed overall engine setup of FRITZ 6 for our self-play matches looked as shown in Table 2.

We executed the self-play matches on several different Windows-98 / NT machines (300 MHz Pentium-II, 333 MHz Celeron, 450 MHz K6-2, 450 MHz & 500 MHz Pentium-III) with 128 MB–256 MB RAM. We used the "Engine Match" function of FRITZ 6 to set up and play the matches at fixed iteration depths for both contestants. The engines with the higher fixed depths always appear as FRITZ 6A in the game scores. By activating the "Alternate Colours" option of the "Engine Match" dialogue, we made sure that all opening positions chosen from the book by random served as sources for two games with the opponents playing reversed colours in each. We used the free CHESSBASE READER program to analyze the generated match databases.

4.1 Calibration Match

We checked the integrity and Black / White fairness of the FRITZ 6 opening book by means of a "calibration" match with 2,500 games between two completely identical opponents. The calibration match pitted FRITZ 6 at a fixed iteration depth of 8 plies against itself. During the calibration match, we disabled the

Table 3. Match details of FRITZ 6 self-play calibration results.

Depth	m	W : D : L / Total	Wins	Draws	Losses	Score
$8_W \Leftrightarrow 8_B$	66	742 : 1,086 : 672 / 2,500	29.68%	43.44%	26.88%	51.40%

Table 4. Statistical analysis of FRITZ 6 self-play calibration results.

Depth	Score	w	s(w)	90%-C Δw	95%-C Δw
$8_W \Leftrightarrow 8_B$	1285.0 / 2,500	0.514	0.010	−0.005, 0.061	−0.011, 0.067

"Alternate Colours" option of the "Engine Match" dialogue in order to avoid useless game repetitions. Thus, the 8-ply FRITZ 6 engines playing Black in the calibration match always appear as FRITZ 6A in the game scores.

Table 3 provides detailed information about the outcome of the calibration match: "m" gives the average number of moves per game and "W : D : L" presents the absolute overall numbers of wins, draws, and losses of the first player ("$8_{W/B}$" denotes 8-ply FRITZ 6 playing White / Black). The remaining columns of the table list the "W : D : L" data and the overall score of the first player as relative percentages of the total game count for the match. Table 4 subjects the results of the calibration match to our procedure of statistical analysis as introduced in Section 3. The extremely level score of the calibration match ("8_W" White 51.4% vs. 48.6% Black "8_B") and its high statistical confidence (just 1% standard error) validate the suitability of both FRITZ 6 and its opening book for our self-play purposes.

Book Calibration. Assuming equal playing strength for FRITZ 6 with Black and White at a fixed depth of 8 plies, the calibration match verifies the integrity and Black / White fairness of the opening book.

Engine Calibration. Assuming the integrity and Black / White fairness of the opening book, the calibration match verifies the equal playing strength of FRITZ 6 with Black and White at a fixed depth of 8 plies. Scaling the engine calibration to other fixed depths requires additional calibration matches at these depths. We did not deem the according effort worthwhile because it is most unlikely that the Black / White behaviour of FRITZ 6 at other fixed depths differs substantially from the one observed by us at 8 plies.

5 Self-Play Results

We executed seven "depth X+1 ⇔ X" handicap matches with FRITZ 6 at fixed iteration depths ranging from 5–12 plies. The overall number of handicap self-play games amounts to 14,650 with 1,050–3,000 games per match. Table 5 provides detailed information about the outcome of the self-play matches (see Section 4.1 and Table 3 for an explanation of the format). Table 6 presents a rigorous statistical analysis of the match results based on the framework introduced in Section 3.

Table 5. Match details of FRITZ 6 self-play results.

Depth	m	W : D : L / Total	Wins	Draws	Losses	Score	ELO
6 ⇔ 5	63	1,686 : 915 : 399 / 3,000	56.20%	30.50%	13.30%	71.45%	+159
7 ⇔ 6	65	1,643 : 1,066 : 291 / 3,000	54.77%	35.53%	9.70%	72.53%	+169
8 ⇔ 7	67	1,457 : 1,212 : 331 / 3,000	48.57%	40.40%	11.03%	68.77%	+137
9 ⇔ 8	66	1,093 : 1,133 : 274 / 2,500	43.72%	45.32%	10.96%	66.38%	+118
10 ⇔ 9	67	434 : 509 : 107 / 1,050	41.33%	48.48%	10.19%	65.57%	+112
11 ⇔ 10	66	404 : 539 : 107 / 1,050	38.48%	51.33%	10.19%	64.14%	+101
12 ⇔ 11	68	375 : 550 : 125 / 1,050	35.71%	52.38%	11.90%	61.90%	+84

Table 6. Statistical analysis of FRITZ 6 self-play results.

Depth	Score	w	s(w)	90%-C Δw	95%-C Δw
6 ⇔ 5	2143.5 / 3,000	0.715	0.008	0.402, 0.456	0.397, 0.461
7 ⇔ 6	2176.0 / 3,000	0.725	0.008	0.424, 0.477	0.419, 0.483
8 ⇔ 7	2063.0 / 3,000	0.688	0.008	0.347, 0.403	0.342, 0.409
9 ⇔ 8	1659.5 / 2,500	0.664	0.009	0.297, 0.359	0.291, 0.365
10 ⇔ 9	688.5 / 1,050	0.656	0.015	0.263, 0.360	0.254, 0.369
11 ⇔ 10	673.5 / 1,050	0.641	0.015	0.234, 0.332	0.225, 0.341
12 ⇔ 11	650.0 / 1,050	0.619	0.015	0.189, 0.287	0.179, 0.297
$[\Delta\mathbf{w}]$	–	–	–	$\emptyset$	$\emptyset$
$[\Delta\mathbf{w}]'$	–	–	–	0.402, 0.403	0.397, 0.409
$[\Delta\mathbf{w}]''$	–	–	–	0.297, 0.332	0.291, 0.297

The stunning conclusion of our experiment is that it not only hints at but clearly shows the existence of diminishing returns for additional search in direct computer self-play by the chess program FRITZ 6. Beyond fixed iteration depths of 7 plies, the scoring rates of the deeper searching program versions steadily decline from 72.5% for "7 ⇔ 6" to a mere 61.9% for "12 ⇔ 11". The experimental data allows us to conclude with 95% statistical confidence that the differences in playing strength of FRITZ 6 in handicap self-play at fixed depths ranging from 9–12 plies are indeed smaller than those at 6–8 plies: $[\Delta w]_{90\%} = [\Delta w]_{95\%} = \emptyset$ and $0.369 = \max(u_{95\%}) < 0.397 = \min(l_{95\%})$ holds for these two sets of depths.

Further evidence for the existence of diminishing returns is visible from the "W : D : L" data. The changes in the rates of games won and drawn by the deeper searching program versions are of particular significance in this context. While the rates of lost games stay fairly constant around 11%, the rates of won games decrease steadily from 56.2% for "6 ⇔ 5" to 35.7% for "12 ⇔ 11" and the rates of drawn games increase reversely from 30.5% for "6 ⇔ 5" to 52.4% for "12 ⇔ 11". Although the deeper searching program versions apparently do not lose more games, they show clear signs of diminishing abilities to win with progressing search depth. Interestingly enough, the average length of the self-play games hardly changes throughout the whole depth range.

Unfortunately, the available data does not allow us to quantify the diminishing returns with good statistical confidence. The non-empty intersections $[\Delta w]'$ for the Δw ranges up to "8 $\Leftrightarrow$ 7"and $[\Delta w]''$ for the remaining ones show that the differences in playing strength could still be constant from 5–8 plies and 8–12 plies respectively. Because of the steadily decreasing scoring rates and Δw bounds, however, we deem this scenario of constants to be highly unlikely.

6 Future Work

In view of our current results laid down in Tables 5 and 6, the course of future self-play research seems quite obvious.

1. Try to quantify the diminishing returns and differences in playing strength at fixed depths up to 12 plies with good statistical confidence $\Rightarrow$ play even more games at these depths.
2. Push the depth range of self-play in computer chess beyond 12 plies and then do the same for these extremely high depths as before.
3. Perform self-play experiments with other chess programs in order to ensure that we are not only measuring weird artifacts of FRITZ 6.

Acknowledgements

ChessBase GmbH donated free copies of FRITZ 6 to us and included the necessary handicap self-play abilities in the program upon our request. Matthias Wüllenweber of ChessBase GmbH proved to be an especially avid supporter of our new self-play experiment. He provided the FRITZ 6A engine clone of the originally released FRITZ 6 version without which the whole experiment would have been impossible.

The availability and exclusive usage of several fast x86-PC compatible machines for a period of several months starting in December 1999 was equally important for the overall success of the experiment. These PCs were kindly provided by the Supertechnologies Group of the Laboratory for Computer Science at the Massachussetts Institute of Technology (M.I.T.), headed by Prof. Leiserson.

References

1. Berliner, H. J. and Goetsch, G. and Campbell, M. S. and Ebeling, C. (1990). Measuring the performance potential of chess programs. *Artificial Intelligence*, Vol. 43, No. 1, pp. 7–21.
2. Brockington, M. G. (1997). KEYANO *unplugged – The construction of an Othello program*. Technical Report TR 97–05, Department of Computing Science, University of Alberta.

3. Condon, J. H. and Thompson, K. (1983). BELLE. *Chess Skill in Man and Machine*, P. W. Frey (ed.), pp. 82–118, Springer, 2nd ed. 1983, ISBN 0-387-90790-4 / 3-540-90790-4.
4. Gillogly. J. J. (1978). *Performance Analysis of the Technology Chess Program*. Ph.D. Thesis, Carnegie-Mellon University [printed as Technical Report CMU-CS-78-189, Computer Science Department, Carnegie-Mellon University].
5. Heinhold, J. and Gaede, K.-W. (1964). *Ingenieur-Statistik*. Oldenbourg, 3rd ed. 1972, ISBN 3-486-31743-1.
6. Heinz, E. A. (2000). *Scalable Search in Computer Chess*. Vieweg / Morgan Kaufmann, ISBN 3-528-05732-7.
7. Heinz, E. A. (2000). Modeling the "go deep" behaviour of CRAFTY and DARKTHOUGHT. *Advances in Computer Games 9*, Proceedings, H. J. van den Herik and B. Monien (eds.), to be published.
8. Heinz, E. A. (2000). Self-play experiments in computer chess revisited. *Advances in Computer Games 9*, Proceedings, H. J. van den Herik and B. Monien (eds.), to be published.
9. Heinz, E. A. (1998). DARKTHOUGHT goes deep. *ICCA Journal*, Vol. 21, No. 4, pp. 228–244.
10. Hyatt, R. M. and Newborn, M. M. (1997). CRAFTY goes deep. *ICCA Journal*, Vol. 20, No. 2, pp. 79–86.
11. Junghanns, A. and Schaeffer, J. (1997). Search versus knowledge in game-playing programs revisited. *15th International Joint Conference on Artificial Intelligence*, Proceedings Vol. I, pp. 692–697, Morgan Kaufmann, ISBN 1-558-60480-4.
12. Junghanns, A. and Schaeffer, J. and Brockington, M. and Björnsson, Y. and Marsland, T. A. (1997). Diminishing returns for additional search in chess. *Advances in Computer Chess 8*, H. J. van den Herik and J. W. H. M. Uiterwijk (eds.), pp. 53–67, University of Maastricht, ISBN 9-062-16234-7.
13. Korf, R. E. (1985). Iterative deepening: An optimal admissible tree search. *Artificial Intelligence*, Vol. 27, No. 1, pp. 97–109.
14. Lee, K.-F. and Mahajan, S. (1990). The development of a world-class Othello program. *Artificial Intelligence*, Vol. 43, No. 1, pp. 21–36.
15. Levy, D. N. L. (1997). Crystal balls: The meta-science of prediction in computer chess. *ICCA Journal*, Vol. 20, No. 2, pp. 71–78.
16. Levy, D. N. L. and Newborn, M. M. (1991). *How Computers Play Chess*. Computer Science Press, ISBN 0-716-78121-2 / 0-716-78239-1.
17. Levy, D. N. L. (1986). When will brute force programs beat Kasparov? *ICCA Journal*, Vol. 9, No. 2, pp. 81–86.
18. Moore, D. S. and McCabe, G. P. (1993). *Introduction to the Practice of Statistics*. W. H. Freyman, 2nd. ed., ISBN 0-716-72250-X.
19. Mysliwietz, P. (1994). *Konstruktion und Optimierung von Bewertungsfunktionen beim Schach*. Dissertation (Ph.D. Thesis), University of Paderborn.
20. Newborn, M. M. (1985). A hypothesis concerning the strength of chess programs. *ICCA Journal*, Vol. 8, No. 4, pp. 209–215.
21. Newborn, M. M. (1979). Recent progress in computer chess. *Advances in Computers*, Vol. 18, pp. 59–117 [reprinted in *Computer Games I*, D. N. L. Levy (ed.), pp. 226–324, Springer, ISBN 0-387-96496-4 / 3-540-96496-4].
22. Newborn, M. M. (1978). Computer chess: Recent progress and future expectations. *3rd Jerusalem Conference on Information Technology*, Proceedings, J. Moneta (ed.), North-Holland, ISBN 0-444-85192-5.
23. Schaeffer, J. (1997). *One Jump Ahead: Challenging Human Supremacy in Checkers*. Springer, ISBN 0-387-94930-5.

24. Schaeffer, J. and Lu, P. and Szafron, D. and Lake, R. (1993). A re-examination of brute-force search, *AAAI Fall Symposium*, Proceedings (AAAI Report FS-93-02: *Intelligent Games – Planning and Learning*), S. Epstein and R. Levinson (eds.), pp. 51–58, AAAI Press, ISBN 0-929-28051-2.
25. Schaeffer, J. (1986). *Experiments in Search and Knowledge*. Ph.D. Thesis, University of Waterloo [reprinted as Technical Report TR 86–12, Department of Computing Science, University of Alberta].
26. Slate, D. J. and Atkin, L.R. (1977). CHESS 4.5 – The Northwestern University chess program. *Chess Skill in Man and Machine*, P.W. Frey (ed.), pp. 82–118, Springer, 2nd ed. 1983, ISBN 0-387-90790-4/3-540-90790-4.
27. Szabo, A. and Szabo, B. (1988). The technology curve revisited. *ICCA Journal*, Vol. 11, No. 1, pp. 14–20.
28. Thompson, K. (1982). Computer chess strength. *Advances in Computer Chess 3*, M. R. B. Clarke (ed.), pp. 55–56, Pergamon, ISBN 0-080-26898-6.