Game Self-Play with Pure Monte-Carlo: The Basin Structure

Ingo Althoefer, Jena University
ingo.althoefer @ uni-jena.de
April 15, 2010

Key words: game search, Monte-Carlo, self-play, diminishing returns, self-play basins.

Abstract

Self-play series are a standard technique in the testing of game programs. X plays against itself with different resources (search depth d vs. d+1, or thinking time t vs. 2t, or n vs. 2n processors). Often, testers observe "rather constant" winning quota for large parameter ranges. Also often, diminishing returns are found at the upper end: The side with larger resources is no longer able to keep the winning quota on the previous high level.

Our new observation: At (very) small resources the winning quota may also be closer to the 50% level. So in total, for the side with fewer resources the performance curve looks like a **basin**. For pure Monte Carlo game search (resource parameter = number of random games per move) we found this **Basin Structure** in games like "Double Step Races", Clobber, ConHex, "Fox versus Hounds", "EinStein wurfelt nicht". For a few games even **double basins** were found. Our very long test series give statistical significance. For one specific game the self-play basin is proved theoretically.

The recognition and understanding of basins may help to improve the automatic evaluation of (new) game rules by very quick self-play series.

1. Introduction

Self-play series are a standard technique in the testing of game programs. Program X plays against itself with different resources (search depth d vs. d+1, or thinking time t vs. 2t, or n vs. 2n processors). Often, testers observe "rather constant" winning quota for large parameter ranges. Also often, diminishing returns are found at the upper end: The side with larger resources is no longer able to keep the winning quota on the previous high level.

In this report we investigate typical structures in self-play series of **pure Monte-Carlo** game search. This most basic Monte-Carlo procedure [Abr 1990] works as follows when used with parameter k: In a given situation S, all feasible moves are determined. For each move, k random games are played. The move with the best score (from the viewpoint of the player who has to move in S) is executed. Ties are broken by fair coin flips. For short, this algorithm is called MC(k). When the game is finite and k goes to infinity, the procedure converges. However, in the limit Monte-Carlo will not necessarily play perfectly according to the game's logic.

Let q(k) be the winning quota of MC(k) vs. MC(2k). Of course, q(k) depends on the game under investigation. In this paper it will always be clear which game is meant. By reasons of fairness, a test series between MC(k) and MC(2k) consists of two equally-weighted parts: In half of the games MC(k) gets the side with the first move, in the other half MC(2k) is in this role. q(k) is the average of the two scores. Normally, q(k) will be between zero and 50 %. By the convergence of MC(k) for k to infinity, q(k) will converge to 50 %.

For many games, we found a structure of the following type: There exists some k* such that
q(1) > q(2) > .. > **q(k*)** < q(k*+1) < ... <
So, at parameter k=k*, the advantage of MC(2k) over MC(k) is most expressed. The sequence of the q-values looks like a basin, with bottom of the basin at k* and borders at k=1 and k=infinity. We call this structure **self-play basin**. A special case is k*=1. In our experiments this borderline case showed up only for a small fraction of the games.

For reasons of economy, often q(k) is not “determined” for all k, but only for the powers of 2 or some other sparse subset of the natural numbers. We also followed this strategy in wide parts of our experimental tests.

What do we mean by the formulation "rather constant" in the first paragraph? Firstly, testers typically have interesting games (like chess, go, Hex, Havannah) in mind, which are more or less complicated, so that self-play games of non-trivial quality take lots of time. Secondly, most testers typically are more interested in high-level or near-perfect play than in almost-random play. So, it is often already a tedious job (costing many CPU-hours or CPU-days) to run series with only a few dozens or a few hundreds games for each pairing. A good impression is got from the experiments collected and described by E.A. Heinz in [Hei 2000].

Now, assume the following q-subsequence for some range (t vs 2t, 2t vs 4t, 4t vs 8t, 8t vs 16t, 16t vs 32t): 41 %, 40 %, 39 %, 40 %, 41 %.
It is a flat basin, with the bottom at 4t-vs-8t. But small test series with only a few hundred games for each pairing are not enough to recognise such a flat basin reliably. Instead, the quota will look like “rather constant with some noise”. A difference of 1 % means one single win/loss-exchange in a series of 100 games. And, the 2-sigma-rule from statistics in mind (see some more on this at the end of Section 2), one knows that a sample of 100 games with expected score-value 40% gives a sample-score outside of the wide [35% , 45%]-interval with probability around 4.5 %. This makes it understandable that short-series testers so often speak of rather constant scores. This in mind, it is not a surprise that so far basin structures had not been identified.

Our focus is on artificial and simple games to allow sufficiently long self-play series. These "dummy" games have not been designed for some commercial game market, but solely as working-horses for laboratory experiments.

The paper is organized as follows. In Section 2 experimental results for several game classes are shown. They all exhibit basin structures. The game Quad-Single-Step-Race-8x9 even exhibits a double basin. Section 3 contains a theoretical proof for a self-play basin of a very simple game. We conclude by a discussion with questions and conjectures in Section 4.

2. Experimental Results on Self-Play Basins

A Class of Abstract Games: Double-Step-Races

Double-step-races were introduced in [Alt 2008]. We repeat the simple rules: Two players, Black and White, move in turn. Black has the first move. Each player has only one stone. This stone is running in a horizontal lane, consisting of finitely many squares. The stone is starting in the leftmost square of its lane. A legal move consists of moving the stone one step or two steps to the right in the lane. The stones of Black and White are running in different lanes. They do not interfere. The player who first reaches the rightmost square in his lane is winner. By historical reasons the race with d+1 squares in each lane is called "Double-Step-Race d". For humans it is obvious that using double steps only (with a single step at the end, if necessary) is an optimal strategy.

Self-play Results for Double-Step-Race-6

Pairing MC(k) vs MC(2k)	Number of games	Score q(k)
1-2	999,999	42.4 %
2-4	**999,999**	**40.9 %**
3-6	999,999	41.0 %
4-8	999,999	41.4 %
5-10	999,999	41.9 %
6-12	999,999	42.3 %
8-16	100,000	43.5 %
16-32	100,000	46.6 %
32-64	100,000	49.4 %
64-128	100,000	50.0 %

q(k) takes its minimum at k=2, so almost at the left border. k=3 gives a narrow runner-up.

Self-play Results for Double-Step-Race-10

Pairing k vs 2k	Number of games	Score q(k)
1-2	999,999	41.6 %
2-4	999,999	40.1 %
4-8	999,999	39.3 %
8-16	**999,999**	**38.8 %**
16-32	999,999	41.6 %
32-64	999,999	46.9 %
64-128	999,999	49.6 %
128-256	999,999	50.0 %

The minimum (amongst the powers of 2) is reached at MC(8) vs. MC(16).

Self-play Results for Double-Step-Race-14

Pairing k vs 2k	Number of games	Score q(k)
1-2	100,000	40.4 %
2-4	100,000	39.0 %
4-8	100,000	37.9 %

8-16	**100,000**	**36.9 %**
16-32	100,000	37.3 %
32-64	100,000	43.0 %
64-128	100,000	48.3 %
128-256	100,000	49.9 %

Like in DSR-10, the bottom is reached at MC(8) vs. MC(16).

Self-play Results for Double-Step-Race-32

Pairing k vs 2k	Number of games	Score q(k)
1-2	10,000	38.7 %
2-4	10,000	35.6 %
4-8	10,000	33.1 %
8-16	999,999	30.9 %
16-32	**100,000**	**30.1 %**
32-64	110,000	30.3 %
64-128	10,000	34.9 %
128-256	10,000	44.8 %

The minimum (amongst the powers of 2) is reached at MC(16) vs. MC(32).

<u>Results for Games which Humans like to play</u>

Clobber on 5x4-Board

Clobber was invented in 2001 by Albert, Grossman, and Nowakowski. Rules and information on first tournaments can be found at [Alt 2002] (computer tournament) and [Gro 2004] (human tournament). Some experiments with pure Monte-Carlo are described in [KSW 2006].

Pairing k vs 2k	Number of games	Score q(k)
1-2	999,999	42.0 %
2-4	**999,999**	**41.8 %**
4-8	700,000	42.5 %
8-16	100,000	43.0 %
16-32	200,000	43.7 %
32-64	189,150	44.6 %
64-128	86,878	44.3 %
128-256	33,387	45.3 %

There is a basin with bottom at 2-4. Longer series will be needed to understand what is happening at 32-64 and 64-128.

Fox and 3 Hounds on 6x6-Board

This is a classical asymmetric game. It is played on the black squares of a chess-style board. All pieces can move only in single steps, and only to free squares. The fox can move in all four diagonal directions (north-east, south-east, south-west, north-west). A hound can move only to the north-east and the north-west. In the beginning the hounds occupy all black squares in the southern back rank. The fox is placed on any other black square. One player commands the fox, the opponent the hounds. The players move in turn, acting with one piece each time. The hounds are to start. The fox is winner when he reaches the southern back rank. The hounds are winners when the fox has become immobile somewhere else. Typically, Fox and 4 Hounds are playing on an 8x8 board. For our self-play experiments we used the smaller 6x6-board with 3 hounds and the fox starting on b2, instead. "Fox and Hounds" is the traditional name of the game. In some sources, like the book [BCG 1982], it is treated under the confusing name "Fox and Geese".

Pairing k vs 2k	Number of games	Score q(k)
1-2	200,000	43.5 %
2-4	200,000	43.2 %
3-6	**200,000**	**43.0 %**
4-8	**200,000**	**43.0 %**
5-10	200,000	43.3 %
6-12	200,000	43.6 %
8-16	200,000	43.9 %
16-32	200,000	45.0 %
400-800	2,000	48.6 %
2000-4000	2,000	50.3 %
4000-8000	2,000	49.8 %

There is a very flat basin for small k, with the bottom likely at 3-6 and/or 4-8. The data for large k, especially the 50+ %-score for 2000-vs-4000 show that 2,000 games are not enough to distinguish reliably values so close to the 50 %-line.

ConHex

This connection game is a modern classic, designed by Michail Antonow [Ant 2002], [Bro 2005].

Pairing k vs 2k	Number of games	Score q(k)
1-2	2,000	37 %
2-4	2,000	32 %
4-8	2,000	27 %
8-16	2,000	23 %
16-32	**2,000**	**21 %**
32-64	2,000	22 %
64-128	2,000	27 %
128-256	2,000	33 %
256-512	2,000	42 %
512-1024	2,000	45 %

2,000 games only for each pairing [Gue 2009] mean rather short self-play series. But the basin is very expressed. Therefore, statistical significance for the basin is given.

EinStein wurfelt nicht

This is a quick game with chance, designed by Althoefer [Alt 2004], [SS 2005].

Pairing k vs 2k	Number of games	Score q(k)
1-2	999,999	43.91 %
2-4	11,335,070	43.54 %

3-6	**9,073,918**	**43.52 %**
4-8	6,365,913	43.62 %
5-10	4,765,254	43.64 %
6-12	1,000,000	43,73 %
8-16	1,966,408	43.82 %
16-32	1,040,226	44.44 %
32-64	10,000	45.8 %
64-128	10,000	47.2 %
128-256	10,000	49.1 %
256-512	10,000	48.9 %

The bottom of the basin is at 3-6, the "central" part of the basin is very flat.. But, clearly the 1-2-score is significantly larger than that of 3-6. At the large parameters, it is likely that the 256-512-score is indeed closer to 50.0 % than that of 128-256. But series with 10,000 games each are too short to decide about that.   For us, it was not a surprise that in this game the q-curve is so flat, as in games with chance (like here in EinStein wurfelt nicht) the chance levels away differences in playing strength to some degree.

A Simple Game with a Self-Play Double Basin:
Quad--Single-Step-Race on 8x9-Board

The Rules: The game is played on a rectangular chess-type board with 8x9 squares, with four white and four black pawns. In the initial position, the white pawns are on the squares a1, b1, c1, d1; the black pawns start on e1, f1, g1, h1. The players move in turn. Each move consists of a single upward-step of a freely chosen own pawn. So, the a-pawn has the lane a1-a2-a3-...-a8-a9 for running up. The e-pawn is on the lane e1-e2-...-e9, and so on. The player who first reaches rank 9 with any of his pawns is winner. For humans, perfect play is trivial: Select any of your pawns and move only with this piece. Pure Monte-Carlo, however, has its problems with the strategy.

Pairing k vs. 2k	Number of games	Score q(k) of MC(k)	Exact Scores q(k) (by T. Fischer)
1-2	8,517,867	35.4%	35.412 %
2-4	**4,886,290**	**34.9%**	**34.930 %**
3-6	999,999	35.5%	35,563 %
4-8	2,686,673	35.7%	35.692 %
5-10	1,771,835	35.5%	35.490 %
6-12	1,492,540	35.2%	35.186 %
7-14	261,532	34.9%	34.884 %
8-16	1,511,461	34.6%	34.620 %
16-32	**750,124**	**33.7%**	**33.732 %**
32-64	10,000	34.7%	34.012 %
64-128	10,000	35.2%	35.341 %

There are two distinct basins, with bottoms at 2-4 and at 16-32. The bottom at 16-32 is the deeper one.

Application of the sigma-calculus from statistics shows that with probability over 99 % indeed $q(2) < q(1)$ for the true q-values; and also with probability over 99 % $q(16) < q(8)$. For instance, the 2-sigma-rule says that in case of n games and expected value E the absolute score X_n

satisfies

$$E - 1/\text{sqrt}(n) < X\_n < E + 1/\text{sqrt}(n)$$

with probability greater than 95.5 %. Here we have exploited the fact that 0 <= E <= 1, and sigma <= 0.5 for all random variables which assume values only from [0,1]. For n = 1,000,000 1/sqrt(n) = 0.1 %. Hence, very likely the true value of q(2) will lie between 34.8 % and 35.0 %. Similarly, the true value of q(16) will be between 33.6 % and 33.8 % with high probability.

We would have preferred to have given the analogous game for classical 8x8 board as a double-basin example. But, there the series of winning quota seems to form only a half and one basins:

Quad--Single-Step-Race on 8x8-Board

Pairing k vs 2k	Number of games	Score q(k)
1-2	**10,000**	**34.7 %**
2-4	10,000	35.2 %
4-8	10,000	36.1 %
8-16	10,000	35.2 %
16-32	**10,000**	**34.3 %**
32-64	10,000	35.4 %
64-128	5,000	37.3 %

The bottom of the half-basin is at k=1, the bottom of the full basin at k=16.

Results on many more games with self-play basins are listed in the online report [Alt 2010]. Also several cases with double basins are included.

3. Theoretical Proof of a Basin

The following games, one for each natural number n > 1, have self-play basins. The structure of their trees is so simple that a short theoretical proof for the basins can be given. The mathematics needed for analysis is mainly the fact that for each real value eps the expression (1 + eps/n) ^ n converges to e ^ eps, for n to infinity. Here, e = 2.718... is Euler's number.

The Rules of the Games

Two players, A and B. At the root, player A is to move. He has the choice between two moves: the first one gives a direct win for him without any more action. The second one leads to a position Z where opponent B is to play. In Z, B has n feasible moves: one of them is a direct win for B, the other (n-1) moves give direct wins for A. That is all.

For any MC-parameter k from 1 on, B will always make the right move in Z. Side A with algorithm MC(k) will go to the inferior position Z only in case of a tie: when all his random games from Z lead to A-wins. This happens with probability (1 - 1/n)^k . In case of a tie, A goes to Z with prob ½ (and to the direct win with the other ½). Now we compose these probabilities for k and for 2k and get the formula

$$q(k) = \tfrac{1}{2} * [1 - \tfrac{1}{2}*(1 - 1/n)^k] + \tfrac{1}{2} * \tfrac{1}{2} * (1 - 1/n)^{2k}$$

The term in the first line comes from the case where MC(k) is A, and the term in the second line from the case where MC(2k) is A.

For large n and very small k, A will get a tie with probability very near to 1. Especially, q(1) is approximately ½ - 1/4n, neglecting a quadratic term in 1/n. So, lim q(1) = 0.5 for n to infinity. Interesting is the range with k=c*n. Substituting (1 – 1/n)^(c*n) by e^(-c) and using standard calculus, the global minimum for q(k) is at k = n/log(2). Here, q(k) is approximately 45.5 %.

Remark: The game above has only two inner nodes: the root with degree 2 and node Z with degree n. This tree may easily be transformed to a tree with the same game-theoretic logic and O(n) inner nodes, such that no node has degree larger than 2.

4. Conclusions and Discussion

Basins are a frequent and natural structure in winning-quota of self-play series, at least for pure Monte-Carlo. Yet, it is not clear which applications the knowledge about the existence and shape of self-play basins will have. In the process of automatic ([Bro 2009], [BM 2010]) or computer-aided [Alt 2003] game inventing self-play series are used extensively. Also, Erdmann with his measurements of chance and skill in games ([Erd 2009], [Erd 2010]) makes intensive use of quick self-play series. In these fields, understanding of self-play basins should be helpful.

We want to distinguish six classes of q-sequences.

“Constant Rate”
For finite games and pure Monte-Carlo this can happen only in the trivial case q(k) = 0.5 for all k.

In all the following classes, MC always has lim q(k) = 0.5 for k to infinity.

"Diminishing Returns“
The q-sequence is monotonically increasing, i.e.
q(1) < q(2) < q(3) < q(4) < …

"Basin"
There is some intermediate parameter k* such that q( ) is monotonically decreasing to the left of k*, and monotonically increasing to the right of k*, i.e:

q(1) > q(2) > …> **q(k*)** < q(k*+1) < q(k*+2) < …

The most natural explanation for this basin structure is that for very small MC-parameters both versions (MC(k) and MC(2k)) have almost no understanding of the game. So, MC(2k) gets only a slight edge by its better local “look-ahead” near the end of the game. Then, for growing k, Monte-Carlo understands the game better and better – especially a factor of 2 for the number of playouts makes a difference. Finally, at large k both versions understand the game almost “perfectly” in MC-sense. This explanation fits well for the races and also for the game from Section 3 with the theoretical analysis.

"A half and one Basins"
There are two intermediate values 1 < m* < k**, such that the q-sequence is monotonically increasing between 1 and m*, monotonically decreasing from m* to k**, and finally monotonically increasing again for k > k**.

"Double Basin"
There are three intermediate values 1 < k* < m* < k**, such that the q-sequence is monotonically decreasing to the left of k*, monotonically increasing between k* and m*, monotonically decreasing again between m* and k**, and finally monotonically increasing again for k > k**.

"More Basins and any other Shapes"
We are firmly convinced that there exist both artificial and normal games with more than two basins. It should be only a question of time and transpiration until such animals are found. Especially, it should be rather easy to construct games with tactical anomalies and corresponding strange q-curves. The special thing with our race games (double-step races as well as quad-single-step-races) is that they have a very monotone structure: in each position Monte-Carlo converges to an optimal move decision when the MC-parameter goes to infinity (formal proof by induction). Nevertheless, these well-behaving games show basins and even double-basins.

In some experiments we determined the q-quota only for powers of 2 (1-2, 2-4, 4-8, 8-16, …). In other cases we had more computing time which allowed us to look also at intermediate pairings like 3-6, 5-10, ... Of course, the bottom of a basin may be at any value of k. One of the reasons to look only at sparse sets like 1, 2, 4, 8, … is that decades ago self-play series started [Tho 1982] from alpha-beta searches where search depth rather than some more detailed "count" is the crucial parameter: For instance, in chess "depth d" means about 5-to-the-power-d leaves which have to be evaluated. Also, in parallel game search the numbers of processors are typically some powers of 2.

Open Questions and Conjectures

Our findings on self-play basins seem to stand in contrast to results from earlier self-play experiments. Especially in chess they seemed to show "rather constant" winning quota for large parameter ranges. Therefore, we ask

Question 1: Are there self-play basins for more advanced game tree algorithms like alpha-beta (with iterative deepening) and UCT [KSW 2006]? So far, no test runs seem to have been conducted for (very) small time parameters.

A very few data points can be interpreted as hints that there may indeed be self-play basins in alpha-beta (for chess) and UCT (for go): In [Haw 2003], Haworth remarked that in one Fritz6 chess experiment conducted by Heinz [Hei 2003], the quota of depth 5 vs depth 6 is too large to fit well into the scheme of "rather constant". At 13x13-go, series played on CGOS show strength for the fastet level of Leela_Lite that is to high [CGS 2008].

Question 2: What does the q-sequence tell about properties of a game? Will "**q-curve reading**" become a standard tool in the process of game evaluation and design?

Conjecture 1: More sophisticated games tend to have broader self-play basins, with the bottom at higher MC-parameters.

Conjecture 2: More sophisticated algorithms (for instance UCT instead of pure Monte-Carlo) should lead to broader self-play basins.

Acknowledgements

Thanks go to Joerg Sameith. He is the author of **McRandom,** a wonderful software for Monte-Carlo self-play experiments. Without McRandom I would never have found the basin structures. Thomas Fischer computed exact q-values for the double-basin example with the help of dynamic programming. The MC self-play data for ConHex are from Joerg Guenther's diploma thesis. Philip Henderson drew my attention to the CGOS website where the curve for Leela_Lite seems to show a slight basin structure. Jakob Erdmann and Thomas Fischer gave constructive feedback on earlier versions of this paper.

References

[Abr 1990] B. Abramson. Expected-Outcome: A General Model of Static Evaluation. IEEE Transactions on Pattern Analysis and Machine Intelligence 12(1990), 182-193.

[Alt 2002] I. Althoefer. Clobber – a new game with very simple rules. ICCA Journal 25 (2002), 123-125.

[Alt 2003] I. Althoefer. Computer-aided game inventing. Technical report, FSU Jena, Fakultaet Mathematik und Informatik, October 2003. Online available at
http://www.minet.uni-jena.de/preprints/althoefer_03/CAGI.pdf

[Alt 2004] I. Althoefer. Board game "EinStein wurfelt nicht", first presented in August 2004.
http://www.boardgamegeek.com/boardgame/18699/einstein-wurfelt-nicht
Rules at http://www.littlegolem.net/jsp/games/gamedetail.jsp?gtid=einstein&page=rules

[Alt 2008] I. Althoefer. On the laziness of Monte-Carlo game tree search in non-tight situations. Technical report, September 2008. Online available at
http://www.althofer.de/mc-laziness.pdf

[Alt 2010] I. Althoefer. Website with self-play results for many games (to be set up in April 2010). http://www.althofer.de/many-self-play-basins.html

[Ant 2002] M. Antonow. Board game "ConHex". Designed and first presented in 2002.
http://www.boardgamegeek.com/boardgame/10989/conhex

[BCG 1982] E.R. Berlekamp, J.H. Conway, and R.K. Guy. Winning ways for your mathematical play, Academic Press, 1982.

[Bro 2005] C. Browne. Connection Games: Variations on a Theme, A. K. Peters, Massachusetts, 2005. (Contains a detailed description of game ConHex.)

[Bro 2009] C. Browne. Automatic Generation and Evaluation of Recombination Games. Ph.D. Thesis, Faculty of Information Technology, Queensland University of Technology, Brisbane, Australia, 2008.

[BM 2010] C. Browne and F. Maire. Evolutionary Game Design. IEEE Transactions on Computational Intelligence and AI in games, 2010.

[CGS 2008] Computer Go Server. Scalability study for 13x13 go, finished April 2008. http://cgos.boardspace.net/study/13/index.html

[Erd 2009] J. Erdmann. Towards a characterization of chance in games - the case of two-player-zero-sum games with perfect information. Technical Report, FSU Jena, Fakultaet Mathematik und Informatik. http://www.minet.uni-jena.de/Math-Net/reports/sources/2009/09-05report.pdf

[Erd 2010] J. Erdmann. The characterization of chance and skill in games. Doctoral dissertation, FSU Jena, Fakultaet Mathematik und Informatik submitted in March 2010.

[FaH 2010] Text on the game "Fox and Hounds". At http://en.wikipedia.org/wiki/Fox_games. Accessed April 13, 2010.

[Gro 2004] J.P. Grossman. Report on the first international Clobber tournament. Theoretical Computer Science 313 (2004), 533-537.

[Gue 2009] J. Guenther. Entwicklung und Erprobung eines starken ConHex-Spielers. Diploma thesis (in German), FSU Jena, Fakultaet Mathematik und Informatik, July 2009. On request available from ingo.althoefer @ uni-jena.de

[Haw 2003] G. Haworth. Self-play: statistical significance. ICGA Journal 26 (2003), 115-118.

[Hei 2000] E. A. Heinz. Scalable Search in Computer Chess. Vieweg, 2000.

[Hei 2003] E.A. Heinz. Follow-up on "self-play, deep search, and diminishing returns". ICGA 26 (2003), 75-80.

[KSW 2006] L. Kocsis, Cs. Szepesvári, and J. Willemson. UCT: Bandit based Monte-Carlo planning in games. Manuscript 2006. Online available at http://www.sztaki.hu/~szcsaba/papers/cg06-ext.pdf

[SS 2005] J. Sameith and S. Schwarz. Strong program "Hanfried" for playing "EinStein wurfelt nicht". Downloadable from http://www.joerg.sameith.net/denken_hanfried.html

[Tho 1982] K. Thompson. Computer chess strength. In "Advances in Computer Chess 3" (Ed. M.R.B. Clarke), Pergamon, 1982, 55-56.