

Table of Contents

[Photos & Games](#)

[Mephisto](#)

[Sun Phoenix](#)

[Strategic Quiescence Search](#)

[Pseudo Code](#)

[Eval+](#)

[History Update](#)

[See also](#)

[Publications](#)

[External Links](#)

[References](#)

[What links here?](#)

[Home](#) * [Engines](#) * **Bobby**

Bobby,

a chess program by [Hans-Joachim Kraas](#) and [Günther Schrüfer](#), competing at various [World Computer-](#) and [World Microcomputer Chess Championships](#) from 1983 until 1995 ^[1]. Bobby had a strong [WCCC 1986](#) in [Cologne, Germany](#), defeating the later Champion [Cray Blitz](#) in round two, with some chances to tie first place until the last round loss against [Schaeffer's Sun Phoenix](#) ^[2]. In 1993, Bobby II won the [3rd International Paderborn Computer Chess Championship](#).

The development of Bobby started in 1982 as part of Kraas' and Schrüfer's MSc thesis. It was written entirely in [Pascal](#) on the [IBM 4341-2](#) ^[3] of the [TU Braunschweig](#). In 1987 Bobby II was a complete redesign in [C](#) on a [Atari ST](#) microcomputer. A commercial successor of Bobby II is the program [Doctor?](#), market by [ChessBase](#) as 16-bit engine in 1995 as analysis engine for [ChessBase/Windows 1.0](#), and in 1998 as 32-bit add-on engine of the [Fritz 5.32](#) package ^[4] ^[5]. According to *Das große Computerschachbuch*. ^[6], Bobby had a sophisticated [evaluation](#), with respect to [king safety](#) and [passed pawns](#). Bobby's search was a STC-Search (solution tree cost oriented search) ^[7], as elaborated in Schrüfer's 1988 PhD thesis ^[8]

Photos & Games

Mephisto



[Günther Schrüfer](#), [Hans-Joachim Kraas](#), [Ossi Weiner](#) and [Richard Lang](#) ^[9]

[WCCC 1986](#), round 4, [Bobby](#) - [Mephisto X](#) ^[10]

```
[Event "WCCC 1986"]
[Site "Cologne, Germany"]
[Date "1986.06.14"]
[Round "4"]
[White "Bobby"]
[Black "Mephisto X"]
[Result "1-0"]
```

```
1. e4 Nf6 2. e5 Nd5 3. d4 d6 4. c4 Nb6 5. exd6 exd6 6. Nc3 Be7 7. h3 O-O
8. Nf3 Nc6 9. Be2 Bf5 10. O-O Qd7 11. Bf4 Rae8 12. a4 Bf6 13. a5 Nc8 14. a6
b6 15. g4 Bxg4 16. hxg4 Qxg4+ 17. Bg3 Nxd4 18. Nxd4 Qxd4 19. Qc2 h6 20
. Rfd1
Qc5 21. Nb5 Be5 22. Rd5 Qc6 23. Bxe5 Rxe5 24. Rxe5 dxe5 25. Bg4 Nd6 26
. Nxa7
Qc5 27. Bd7 e4 28. Nb5 Nxb5 29. Bxb5 Qg5+ 30. Kf1 Qh4 31. Rd1 Qh1+ 32.
Ke2
Qf3+ 33. Ke1 Qh1+ 34. Kd2 Qg2 35. Kc1 Qg6 36. Qb3 c6 37. Ba4 c5 38. Qg
3 Ra8
39. Qxg6 fxg6 40. Bb5 g5 41. Rd7 h5 42. a7 e3 43. fxe3 Rf8 44. Bc6 h4
45. a8=Q Rxa8 46. Bxa8 h3 47. Rd5 1-0
```

Sun Phoenix



[WCCC 1986](#), round 5, [Vlastimil Hort](#), [Frans Morsch](#), [Hans van der Zijden](#), [Hans-Joachim Kraas](#) ^[11]

[WCCC 1986](#), round 5, [Bobby](#) - [Sun Phoenix](#)

```
[Event "WCCC 1986"]
[Site "Cologne, Germany"]
[Date "1986.06.15"]
[Round "5"]
[White "Bobby"]
[Black "Sun Phoenix"]
[Result "0-1"]
```

```
1. e4 e6 2. b3 d5 3. Bb2 dxe4 4. Nc3 Nf6 5. Qe2 Be7 6. O-O-
O Qd4 7. Re1 O-O
8. Nxe4 Qxe4 9. Qxe4 Nxe4 10. Rxe4 Nd7 11. Nf3 Bc5 12. d4 Nf6 13. Rh4
Be7
14. Kbl e5 15. h3 e4 16. Ne5 Be6 17. Bc4 Bxc4 18. bxc4 c5 19. d5 e3 20
. Rf4
exf2 21. Rxf2 Ne4 22. Rf4 Nd2+ 23. Ka1 Bd6 24. Re1 Rae8 25. Re2 f5 26.
Rh4 Ne4
27. Nd3 Re7 28. Re3 g6 29. Nc1 Rfe8 30. a4 h5 31. Nb3 Kh7 32. Na5 Kh6
33. Rf3
Ng3 34. Rb3 b6 35. Nc6 Re1+ 36. Ka2 g5 37. Nxa7 Ra8 38. Nb5 Rxa4+ 39.
Ba3 Be5
40. Nd4 0-1
```

Strategic Quiescence Search

Bobby applied a so called *Strategic Quiescence Search* as described in a 1989 [ICCA Journal](#) paper by [Günther Schrüfer](#) ^[12]. Schrüfer claimed the inclusion of certain [none tactical moves](#) in [quiescence search](#) an improvement over a pure tactical search in Bobby. He also stated, for some programs it might be a significant speed penalty in the number of [moves searched per second](#) to do so, i.e. in [generating](#) quiet

moves. In Bobby this was irrelevant, since many of the values required have been precomputed for each node in any case. Beside the standard [standing pat forward pruning](#) conditions FP1 and FP2, all successors, not only [captures](#), [promotions](#) and [checks](#), were tested versus the forward pruning condition FP3.

Pseudo Code

```
int StategicQS(CNode n, int  $\alpha$ , int  $\beta$ ) {
    int bestval = eval(n);
    if ( bestval >=  $\beta$  ) return bestval;    // FP1, fail soft
    if ( bestval >  $\alpha$  )  $\alpha$  = bestval;    // FP2
    for all  $n' \in \text{SUCC}(n)$  do {          // search loop
        if ( evalPlus( $n'$ ) >  $\alpha$  ) {      // FP3
            int actual = -StategicQS( $n'$ , - $\beta$ , - $\alpha$ );
            if ( actual > bestval ) {
                bestval = actual;
                if ( bestval >=  $\beta$  ) return bestval;
                if ( bestval >  $\alpha$  )  $\alpha$  = bestval;
            }
        }
        else if ( evalPlus( $n'$ ) > bestval ) bestval = evalPlus( $n'$ );
    }
    return bestval;
}
```

Eval+

The *evalPlus* value is + ∞ in case of checking moves near the horizon, but scaled to zero for deeper searches to avoid "infinite" checks and [search explosion](#). Otherwise *evalPlus* is [incrementally calculated](#) by *eval(n)* and [move](#) properties. In case of [tactical moves](#), the sum of *eval(n)* and the value of a captured and/or promoted piece and a constant representing half the value of a Pawn is taken. For [quiet](#) or strategical moves, *evalPlus* relies on maximum [score](#) differences of two consecutive evaluations n' and n , triggered by [history success counters](#):

```
if ( isCheck )
    return infiniteIfNearBelowHorizon(depth);
if ( isCapture && isPromotion )
    return eval(n) - VALUE_PAWN/2 + Value(captured piece) + Value(
promoted piece);
if ( isCapture )
    return eval(n) + VALUE_PAWN/2 + Value(captured piece);
if ( isPromotion )
    return eval(n) - VALUE_PAWN/2 + Value(promoted piece);
return eval(n) + hist[from][to].diff;
```

History Update

Positive history based success counters are associated with the maximum difference of two consecutive evaluations n' and n found so far, while zero saturated counters also have zero difference and are therefore always pruned by FP3. Following scheme is used to update the [butterfly boards](#):

```
for all  $n' \in \text{SUCC}(n)$  do {
    int actual = -search( $n'$ , - $\beta$ , -?);
    ...
    if ( none tactical ) {
        // History update by quiet move  $n \rightarrow n'$ 
        if ( eval( $n'$ ) > eval( $n$ ) && actual > eval( $n$ ) ) {
            hist[from][to].success += 10;
            if ( eval( $n'$ ) - eval( $n$ ) > hist[from][to].diff )
                hist[from][to].diff = eval( $n'$ ) - eval( $n$ );
        }
        else if ( entry.success > 0 ) {
            hist[from][to].success--;
            if ( hist[from][to].success == 0 ) hist[from][to].diff =
0;
        }
    }
    ...
}
```

See also

- [Chess legends](#)
- [Fischerle](#)

Publications

- [Rainer Bartel](#), [Hans-Joachim Kraas](#), [Günther Schröder](#) (1985). *Das große Computerschachbuch*. [Data Becker](#), Düsseldorf, ISBN 3-89011-117-3 (German), from [amazon.de](#)
- [Günther Schröder](#) (1988). [Minimax-Suchen : Kosten, Qualität und Algorithmen](#). Braunschweig : Technische Universität. (German)
- [Günther Schröder](#) (1989). *A Strategic Quiescence Search*. [ICCA Journal](#), Vol. 12, No. 1, pp. 3-9.

External Links

- [Bobby's ICGA Tournaments](#)

- [Bobby \(disambiguation page\) from Wikipedia](#)
- [Bobby Fischer from Wikipedia](#)

References

1. [^ Bobby's ICGA Tournaments](#)
2. [^ Helmut Horacek](#), (1986). *The Fifth World Computer Chess Championship Cologne, 1986*, Research Unit for Information Science and Artificial Intelligence, [University of Hamburg](#), from *Kings move Welcome to the 1989 AGT World Computer Chess Championship*. pg 21, available as [pdf reprint](#) from [The Computer History Museum](#)
3. [^ IBM Reference / Glossary](#)
4. [^ Fritz 5.32 - mehr als nur ein Update!](#) by [Peter Schreiner](#) (German), [Schachclub Leinzell](#)
5. [^ Doctor? 2.0 / Engine MacIntosh](#) from [Schachversand Niggemann](#)
6. [^ Rainer Bartel](#), [Hans-Joachim Kraas](#), [Günther Schrüfer](#) (1985). *Das große Computerschachbuch*. [Data Becker](#), Düsseldorf, ISBN 3-89011-117-3 (German), from [amazon.de](#)
7. [^ Doctor? by Dr. Hans-Joachim Kraas and Dr. Gunther Schrüfer](#)
8. [^ Günther Schrüfer](#) (1988). [Minimax-Suchen : Kosten, Qualität und Algorithmen](#). Braunschweig : Technische Universität. (German)
9. [^ László Lindner](#), A SZÁMÍTÓGÉPES SAKK KÉPEKBEN című melléklete - The pictures of the Beginning of Chess Computers
10. [^ Cologne 1986, Chess, Round 4, Game 4](#)
11. [^ Image](#) from [Pierre Nolot](#) (1986). *Echecs: Les Progès des Programmes*. [Jeux et Stratégie](#)
12. [^ Günther Schrüfer](#) (1989). *A Strategic Quiescence Search*. [ICCA Journal](#), Vol. 12, No. 1, pp. 3-9

What links here?

Page	Date Edited
ACM 1985	Jun 21, 2017
AMDAHL 470	Jul 19, 2016
Bernie	Oct 31, 2015
Bobby	Jan 7, 2016
Butterfly Boards	Jan 27, 2015
Doctor?	Jun 7, 2016
Engines	Mar 10, 2018
Fischerle	Aug 21, 2017
Günther Schrüfer	Jan 7, 2016
Hans-Joachim Kraas	Jan 7, 2016
History Leaf Pruning	Apr 15, 2013
IBM 370	Jan 20, 2018
IPCCC	Jan 20, 2017
IPCCC 1991	Jun 9, 2016
IPCCC 1993	Nov 1, 2016
IPCCC 1994	Jul 31, 2013
King Pawn Tropism	Sep 1, 2014
Late Move Reductions	Sep 25, 2017

Page	Date Edited
Nona	Jan 7, 2016
Pruning	Jan 1, 2018
Quiescence Search	Aug 19, 2017
Relative History Heuristic	Jun 8, 2015
Selectivity	Jan 12, 2018
Shy	Jul 19, 2016
Strategy	Jan 18, 2018
WCCC 1983	Jan 20, 2018
WCCC 1986	Jul 27, 2017
WMCCC 1993	Dec 30, 2017
WMCCC 1995	Jan 3, 2018

[Up one Level](#)