

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



Phoenix depicted by [F.J. Bertuch](#)
[\[3\]](#)

Phoenix, (Sun Phoenix)
a chess program by [Jonathan Schaeffer](#), which in 1982 rose from the ashes of [Prodigy](#) [\[1\]](#). In the later 80s [Marius Olafsson](#) helped with the implementation of a [parallel search](#). Phoenix and Sun Phoenix participated at four [World Computer Chess Championships](#), the [WCCC 1983](#), [WCCC 1986](#), [WCCC 1989](#), and [WCCC 1995](#). It tied for first place in the 1986 World Championships. Phoenix further played seven consecutive [ACM North American Computer Chess Championships](#), beside the mentioned [WCCC 1983](#) simultaneously the ACM 1983, the [ACM 1984](#), [ACM 1985](#), [ACM 1986](#), [ACM 1987](#), [ACM 1988](#), and [ACM 1989](#).

In his 1987 [ICCA Journal](#) paper *Speculative Computing*, Schaeffer mentions [The Null-Move Algorithm](#) or [Don Beal's null-move](#), and used it none [recursively](#) up to once per search path in his tactical [scout](#) solver *Minix* (Mini-Phoenix), which up and then gave the parallel running Phoenix, which was a less deep searcher than *Minix*, some tactical hints [\[2\]](#).

Table of Contents

[Authors](#)

[Namesake](#)

[See also](#)

[Quotes](#)

[Waterloo](#)

[Zobrist Hashing](#)

[WCCC 1986](#)

[Descriptions](#)

[1989](#)

[1995](#)

[Selected Games](#)

[Publications](#)

[Forum Posts](#)

[External Links](#)

[Chess Program](#)

[Phoenix Chess Systems](#)

[Phoenix elsewhere](#)

[References](#)

[What links here?](#)

Authors

- [Jonathan Schaeffer](#)
- [Marius Olafsson](#)

Namesake

- [Phoenix \(Rahul\)](#)

See also

- [Bird](#)
- [Demonology](#)
- [Mythology](#)
- [Planner](#)
- [Prodigy](#)

Quotes

[Waterloo](#)

[Jonathan Schaeffer](#) in [One Jump Ahead](#), pp. 8 ^[4]:

My Ph.D. was not going well, so in the summer of 1982 I started looking for a distraction. Yes, I started writing yet another chess program, this one called Phoenix (it rose from the ashes of [Prodigy](#)). The [Planner](#) and Prodigy experiences were invaluable, as they convinced me that contrary to all my expectations, lots of chess knowledge didn't work. Which programs were winning the tournaments? The ones with little knowledge, but with the ability to consider an enormous number of chess positions. With a twinge to regret, I wrote Phoenix to mimic these "dumb" programs. The results were immediate. Phoenix didn't know nearly as much about chess as Prodigy did, but it would continually beat it game after game. Obviously, my old approach, imparting human knowledge to an inanimate machine, wasn't the best way to train a computer to play strong chess.

[Zobrist Hashing](#)

[Jonathan Schaeffer](#) on [Zobrist Hashing](#) ^[5]:

... I can speak from experience here. In the early versions of my chess program Phoenix, I generated my Zobrist hash numbers using my student id number as a seed, naively thinking the [random numbers generated](#) by this seed would be good enough. A few years later I put code in to detect when my 32-bit hash key matched the wrong position. To my surprise, there were **lots** of errors. I changed my seed to another number and the error rate dropped dramatically. With this better seed, it became very, very rare to see a hash error. All randomly generated numbers are not the same!

[WCCC 1986](#)

[Jonathan Schaeffer](#) in [One Jump Ahead](#), pp. 9 ^[6]:

I worked hard on Phoenix in preparation of the triennial [World Computer Chess Championship in 1986](#). To improve the program's performance it was modified to run in [parallel](#), using up to thirty computers. They would divide up the work, and each computer would solve part of the problem. In effect, the program was like a small business organization, with a manager to allocate work and employees to do the assigned tasks. The hard work payed off, Phoenix tied for first place in the world championship. I partied late into the night after the final game, intoxicated with success and Coca-Cola. It took a long time for me to come down from my high.

Descriptions

1989

given in the [WCCC 1989](#) booklet ^[7]:

Phoenix uses state-of-the-art search techniques including [singular extensions](#), [minimal window](#) searching, [transposition tables](#), and the [history heuristic](#). The program has lots of chess knowledge, including an extensive long range planner.

Phoenix is capable of running in parallel on a network of machines. In tournament mode, Phoenix is actually two programs: ParaPhoenix and ParaMinix. ParaPhoenix uses 10 [Sun-4s](#) to build trees looking for the best positional move. ParaMinix uses 10 Sun 4s to build trees looking for the best tactical moves. Because of its specialized task, ParaMinix is capable of searching 1-2 ply deeper than ParaPhoenix. ParaMinix has the ability to veto ParaPhoenix's move choice, if a tactically superior move is found. The parallelism is achieved using the [Dynamic Principal Variation Splitting Algorithm](#).

1995

Description given in [1995](#) from the [ICGA](#) site ^[8]:

The Phoenix program was an active participant in the 1980's computer chess tournaments and tied for first place in the 1986 World Championships. The program competing this year is essentially the same as that which competed in the 1989 World Computer Chess Championships. Phoenix's participation in the 1995 championships will serve as a benchmark for measuring improvements in the field. Phoenix will be running on an [HP 9000/720](#) with 64MB [RAM](#), which will be comparable to

that used by most participants, and therefore the primary difference will be in the software. Expectations are that the software advances in the last 6 years will allow the other programs to move past Phoenix '89 in the final standings. Perhaps the best possible outcome would have Phoenix finishing in last place, providing some experimental evidence of the progress in the field!

Selected Games

[WCCC 1986](#), round 3, [BCP](#) - [Sun Phoenix](#) ^[9]

```
[Event "WCCC 1986"]
[Site "Cologne, Germany"]
[Date "1986.06.13"]
[Round "3"]
[White "BCP"]
[Black "Sun Phoenix"]
[Result "0-1"]
```

```
1.e4 e6 2.d4 d5 3.e5 c5 4.c3 Qb6 5.Nf3 Bd7 6.Bd3 cxd4 7.Nxd4 Nc6 8.Nxc
6 Bxc6
9.O-O O-O-
O 10.Nd2 f6 11.Qg4 Re8 12.Re1 Nh6 13.Qh3 Bc5 14.Qg3 Ng4 15.Qxg4 Bxf2+
16.Kf1 Bxe1 17.Kxe1 fxe5 18.Qxg7 Qe3+ 19.Be2 Rhg8 20.Qf7 Bb5 21.Qf2 Qx
e2+
22.Qxe2 Bxe2 23.Kxe2 Rxg2+ 24.Ke3 Rxh2 25.a4 Rf8 26.a5 h5 27.Ra4 Rh1 2
8.Ra1 h4
29.Rb1 Re1+ 30.Kd3 h3 31.Nf1 Rfxf1 0-1
```

Publications

- [Jonathan Schaeffer](#), [Tony Marsland](#) (1985). The Utility of Expert Knowledge. Proceedings [IJCAI 85](#), pp. 585-587. Los Angeles. [pdf](#)
- [Jonathan Schaeffer](#) (1986). *Improved Parallel Alpha-Beta Searching*. Proceedings ACM/IEEE Fall Joint Computer Conference, pp. 519-527.
- [Jonathan Schaeffer](#) (1987). *Speculative Computing*. [ICCA Journal, Vol. 10, No. 3](#)
- [Jonathan Schaeffer](#) (1989). *Distributed Game-Tree Searching*. Journal of Parallel and Distributed Computing, Vol. 6, No. 2, pp. 90-114. ISSN 0743-7315.

Forum Posts

- [QMW computer chess](#) by [Don Beal](#), [rec.games.chess](#), August 19, 1993 » [UPCCC 1993](#)

External Links

Chess Program

- [Phoenix' ICGA Tournaments](#)
- [Computer Chess and Canada](#) by [David Cohen](#), 2005

Phoenix Chess Systems

- [Phoenix Chess Systems](#) from [schach-computer.info](#)
- [Welcome to Phoenix Chess Systems](#) by [Ruud Martin](#)

Phoenix elsewhere

- [Phoenix \(mythology\)](#) from Wikipedia
- [Phoenix \(demonology\)](#) from Wikipedia
- [phoenix: a world of chess](#), Zurich Chess Club (German)
- [Phoenix \(band\)](#) from Wikipedia
- [Hans Koller Free Sound](#) - Nicolas 1-2, [Phoenix](#), 1973, [YouTube](#) Video
line-up: [Adelhard Roidinger](#), [Albert Mair](#), [Alex Bally](#), [Hans Koller](#), [Jürgen Wuchner](#)

References

1. [^] [Jonathan Schaeffer](#) (1997, 2009). [One Jump Ahead](#). 1. This Was Going to Be Easy, pp. 8
2. [^] [Jonathan Schaeffer](#) (1987). *Speculative Computing*. [ICCA Journal, Vol. 10, No. 3](#)
3. [^] [Phoenix \(mythology\)](#) from Wikipedia
4. [^] [Jonathan Schaeffer](#) (1997, 2009). [One Jump Ahead](#). 1. This Was Going to Be Easy, pp. 8
5. [^] [Hash tables - Clash!!! What happens next?](#) by [Valavan Manohararajah](#), [rec.games.chess](#), post 6 by [Jonathan Schaeffer](#), March 17, 1994
6. [^] [Jonathan Schaeffer](#) (1997, 2009). [One Jump Ahead](#). 1. This Was Going to Be Easy, pp. 9
7. [^] [Kings Move - Welcome to the 1989 AGT World Computer Chess Championship](#). Edmonton, Alberta, Canada, Courtesy of [Peter Jennings](#), from [The Computer History Museum](#), [pdf](#)
8. [^] [Phoenix's ICGA Tournaments](#)
9. [^] [Cologne 1986 - Chess - Round 3 - Game 6 \(ICGA Tournaments\)](#)

What links here?

Page

[ACM 1984](#)

[ACM 1985](#)

Date Edited

Jul 19, 2016

Jun 21, 2017

Page	Date Edited
ACM 1986	Dec 5, 2017
ACM 1987	Dec 5, 2017
ACM 1988	Dec 5, 2017
ACM 1989	Jun 7, 2016
Aegon 1988	Jan 22, 2017
Awit	Dec 22, 2017
Bobby	Jan 7, 2016
Canadian Computer Chess Invitational Championship 1984	Mar 2, 2013
ChipTest	Jun 19, 2016
Engines	Mar 10, 2018
Fire	Oct 12, 2017
Honeywell 6000	Jan 19, 2018
Jonathan Schaeffer	Jan 29, 2018
Marius Olafsson	Jan 9, 2016
Mathematician	Apr 9, 2018
Mephisto Almeria	Sep 1, 2017
Michael George	Jan 3, 2016
Netherlands-vs-Computers-1989	Mar 10, 2017
Nona	Jan 7, 2016
Null Move Pruning	Dec 2, 2017
Phoenix	May 8, 2017
Phoenix (Rahul)	Mar 31, 2016
Planner	Sep 24, 2012
Prodigy	Jan 19, 2018
Shy	Jul 19, 2016
SPOC	Jun 21, 2017
Sun	Jun 7, 2016
University of Alberta	Feb 20, 2018
University of Waterloo	Jan 29, 2018
UPCCC 1992	Mar 9, 2012
UPCCC 1993	Mar 6, 2015
UPCCC 1994	Mar 9, 2012
WCCC 1983	Jan 20, 2018
WCCC 1986	Jul 27, 2017
WCCC 1989	Dec 5, 2017
WCCC 1995	Jul 12, 2017
Young Brothers Wait Concept	Apr 22, 2018
Zobrist Hashing	Jan 22, 2018

[Up one level](#)