

Table of Contents

[Descriptions](#)

[ACM](#)

[Man-Machine Studies](#)

[Mate in one?](#)

[Mater](#)

[See also](#)

[Publications](#)

[External Links](#)

[References](#)

[What links here?](#)

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

Coko,

a chess program by [Dennis Cooper](#) and [Ed Kozdrowicki](#) which competed the first four [ACM North American Computer Chess Championships](#), [ACM 1970](#), [ACM 1971](#), [ACM 1972](#) (Coko III) and [ACM 1972](#) (Coko IV). Coko, the Cooper-Kozdrowicki chess program was written in [Fortran](#) as a highly selective tree searcher in the spirit of a [Shannon Type B](#) program using a tree pruning system (TPS) consists of a set of commands designed for programming heuristic tree searches ^[1]. As described by Cooper in the 1971 Panel ^[2], Coko III does not use [Alpha-Beta](#):

Although the alpha-beta procedure is a great time saving method, it is unclear at this stage of program development what the full significance of applying such a method to a tactical-strategical game tree would be. Coko III does save the chess tree with periodic pruning to allow for the addition of more branches.

Descriptions

ACM

[Communications of the ACM](#), Vol. 16, 7 ^[3]

COKO III is a chess player written entirely in Fortran. On the [IBM 360-65](#), COKO III plays a minimal chess game at the rate of .2 sec cpu time per move, with a level close to lower chess club play. A selective tree searching procedure controlled by tactical chess logistics allows a deployment of multiple minimal game calculations to achieve some optimal move selection. The tree searching algorithms are the heart of COKO's effectiveness, yet they are conceptually simple. In addition, an interesting phenomenon called a tree searching catastrophe has plagued COKO's entire development just as it troubles a human player. Standard exponential growth is curbed to a large extent by the definition and trimming of the Fisher set. A clear distinction between tree pruning and selective tree searching is also made. Representation of the chess environment is described along with a strategical preanalysis procedure that maps the Lasker regions. Specific chess algorithms are described which could be used as a command structure by anyone desiring to do some chess program experimentation. A comparison is made of some mysterious actions of human players and COKO III.

Man-Machine Studies

International Journal of Man-Machine Studies ^[4]

The performance capabilities of the best computer chess programs are compared with their human counterparts with emphasis being placed on machine behavior limits. A grandmaster usually spends a lifetime collecting knowledge or information about the game. Some of this knowledge is given to COKO in the form of a 12 000-line FORTRAN program. Using this knowledge COKO plays very poorly but at the super rate of approximately one move/see. The use of a brute-force selective tree-searching procedure yields an order of magnitude improvement in performance at the standard rate of 3 min/move. Perhaps three orders of magnitude additional improvement is needed to defeat the world champion, a gap which must be bridged, if ever, by programming more chess knowledge into the machine. This paper discusses the "tree-searching catastrophe" as a natural phenomenon that plagues selective tree searching for both man and machine. In addition so-called "interminimal-game communication" is considered as a natural, powerful procedure frequently used by humans to guide their selective search and as a point of emphasis for future development. It is concluded that COKO's development is just beginning, with no immediate barriers to progress, and no lack of ideas for improvement. At present COKO combines brilliant solutions to individual board position puzzles with unimaginable blunders.

Mate in one?

During the [ACM 1971](#), Coko III offered a pawn versus [Genie](#) which, if accepted, would permit a mating sequence 17 plies deep. The pawn was taken, mate was announced, and the predicted line was followed, until ... ^[5]

```
8/pp6/2p2p2/6p1/1P6/2Q1P3/k1K2PPP/5B1R w - - 0 38
```

Apparently due to a [bug](#), Coko III found other moves better than mate in one and threw the win ...

```
38. Kc1 f5 39. Kc2 f4 40. Kc1 g4 41. Kc2 f3 42. Kc1 fxg2 43. Kc2 gxh1=
Q 44. Kc1 Qxf1+
and resigned later 0-1
```

Mater

According to [Monroe Newborn](#) in 1975, the chess [mating combinations](#) program [Mater](#) by [George Baylor](#) and [Herbert Simon](#), initially written in [IPL V](#), was ported to [Fortran](#) and incorporated into Coko ^[6]:
MATER is written by George Baylor and Simon in FORTRAN. It is able to search to great depths for checkmates. MATER is presently part of the Cooper-Kozdrowicki program. While MATER is an interesting program in its own right, the opportunity to checkmate one's opponent plays a relatively small computational part of the game of chess, and its inclusion in the Cooper-Kozdrowicki program does not seem to add measurably to the program's strength.

See also

- [Acronym](#)

Publications

- [Edward W. Kozdrowicki](#), [John S. Licwinko](#), [Dennis W. Cooper](#) (1971). [Algorithms for a minimal chess player: A blitz player](#). [International Journal of Man-Machine Studies](#), Vol 3, 2
- [Edward W. Kozdrowicki](#), [Dennis W. Cooper](#) (1973). [COKO III: the Cooper-Koz chess program](#), [Communications of the ACM](#), Vol. 16, 7
- [Edward W. Kozdrowicki](#), [Dennis W. Cooper](#) (1973). [COKO III and the future of inter-snap](#)

[judgment communication](#). Proceedings of the ACM annual conference

- [Paul Rushton](#), [Tony Marsland](#) (1973). *Current Chess Programs: A Summary of their Potential and Limitations*. INFOR Journal of the Canadian Information Processing Society Vol. 11, No. 1, [pdf](#)
- [Edward W. Kozdrowicki](#), [Dennis W. Cooper](#) (1974). [COKO III: The Cooper-Kozdrowicki chess program](#). [International Journal of Man-Machine Studies](#) Vol. 6, 6

External Links

- [ACM COMPUTER CHESS](#) by [Bill Wall](#)
- [Classic Computer Chess - ... The programs of yesteryear](#) by [Carey](#), hosted by the [Internet Archive](#)

References

1. [^](#) [Edward W. Kozdrowicki](#) (1968). [An adaptive tree pruning system: A language for programming heuristic tree searches](#). Proceedings of the 1968 23rd ACM national conference
2. [^](#) [Ben Mittman](#) (1971). [Computer Chess Programs \(Panel\)](#). [pdf](#) from [The Computer History Museum](#)
3. [^](#) [Edward W. Kozdrowicki](#), [Dennis W. Cooper](#) (1973). [COKO III: the Cooper-Koz chess program](#). [Communications of the ACM](#), Vol. 16, 7
4. [^](#) [Edward W. Kozdrowicki](#), [Dennis W. Cooper](#) (1974). [COKO III: The Cooper-Kozdrowicki chess program](#). [International Journal of Man-Machine Studies](#) Vol. 6, 6
5. [^](#) [George Atkinson](#) (1998). [Chess and Machine Intuition](#). (Intellect Ltd.) pp 63
6. [^](#) [Monroe Newborn](#) (1975). *Computer Chess*. Academic Press, New York, N.Y. p. 26
7. [^](#) [Re: Old programs CHAOS and USC](#) by [Dann Corbit](#), [CCC](#), July 11, 2015

What links here?

Page	Date Edited
0x88	Nov 28, 2016
ACM 1970	Jun 13, 2015
ACM 1971	Jul 19, 2016
ACM 1972	Dec 22, 2017
ACM 1973	Jan 19, 2018
Coko	Jul 13, 2015
Daly CP	Jul 13, 2015
Dennis Cooper	Aug 18, 2014
Ed Kozdrowicki	Aug 18, 2014
Engine Testing	Feb 1, 2018
Engines	Mar 10, 2018
Genie	Feb 7, 2016
IBM 360	Jan 2, 2016
J. Biit	Jan 17, 2017
Jacques Dutka	Jul 7, 2012
Keith Gorlen	May 23, 2016

Page	Date Edited
Mater	Apr 26, 2016
MSU CP	Jul 19, 2016
Paul Rushton	Jan 12, 2015
SOMA	Mar 25, 2015
Steven M. Bellovin	Dec 14, 2017
Tech	Apr 20, 2018
UNIVAC 1100	Jul 21, 2016

[Up one level](#)