LOGISTELLO

is one of today's strongest Othello programs. Besides the powerful hardware on which it is running and the efficient implementation of standard game-tree searching techniques, the program's considerable playing strength is mainly due to several new approaches for the construction of evaluation features, their combination, selective search, and learning from previous games which I have investigated in my Ph.D. thesis and considerably improved while working at NECI.

Download Logistello's book skeleton (i.e. (self-)played games) for studying openings, training evaluation parameters, or just improving your bot's book. Games are stored as ASCII move sequences annotated with the final game result in black's view. All ~37K lines are at least 24-ply WLD correct. The file is gzip'ed. Click here if you prefer the WTHOR format.

Logistello's source code is now available. Last update: November/4/2002: Stephane Nicolet found a bug in the MPC code - fixed.
Logistello's X-window GUI source code.

README file:

This is the Logistello source directory which is licensed under the GPL. I worked on it from 1992 till 1998. As a result, it's not a pretty sight: a zillion tools for handling different game file formats, computing statistics, solving logistic and linear regression systems, book learning, running tournaments etc. Many of them are now obsolete. Some of the code is hard to read - some comments and variable names are even in German (Brett=board,Zug=move,Wert=value). I hardly can tell which file is actually used anymore. The makefiles can be a starting point for exploring the source code. The current Logistello version is generated by "make ../oplayl". I have not included the training data, nor the opening book, nor the pattern and Multi-ProbCut tables. Although the program compiles, it cannot be used to play Othello out of the box. All the software for creating missing tables is included, however. What's missing is the 17 million training positions - which are not that hard to generate nowadays.

The purpose of this release is to present the implementation details of a strong game-playing program. My hope is that it is useful for designing coming generations of even stronger programs.

- Michael Buro, November 2002.

Program overview

- Evaluation features
  - game stage dependent tables for each of the following patterns:
    - horizontals/verticals of length 8
    - diagonals of length 4-8
    - 3x3 corner
    - 2x5 corner
    - edge+2X
  - a simple parity measure
- Feature combination
  - linear
- Search
  - NegaScout with corner quiescence search and multi-ProbCut
  - iterative deepening
- Move sorting
  - hash-table containing moves and value bounds (2**21 entries à 8 bytes)

    - response killer lists
    - shallow searches
- Search speed (on a Pentium-Pro 200 (1997), on today's hardware much faster)
    - middle-game: ~160,000 nodes/sec
    - endgame: ~480,000 nodes/sec
- Search depth (in a 2x30 minutes game)
    - middle-game: 18-23 selective including 10-15 brute-force ply
    - endgame: win/loss/draw determination at 26-22 empty squares, exact score 1-2 ply later
- Opening book
    - consists at the moment of ca. 23k games and evaluations of "best" move alternatives
    - is automatically updated
    - currently several machines are working all day long on book improvement
- Miscellany
    - thinking on opponent's time
    - communication with IOS via socket or with graphical interface via file system
- OS / Language / Compiler
    - Unix / C / gcc 2.7.2.1

Development History

- 1991: first experiments with statistical feature combination
- 1992: searching for significant and fast evaluation features
- 1993: here they are: estimated pattern tables, line approximation of mobility and potential mobility; features are combined using logistic regression; L wins its first tournament ("Paderborn I") achieving 16k nodes/sec in middle-game on a SPARC-10/M20.
- 1994: found an effective selective search approach: ProbCut; learning opening book
- 1995: 10-disc patterns; new book algo. which maximizes out-of-book evaluation
- 1996: multi-ProbCut; hashtable improvement
- 1997: new evaluation function based on plain linear regression; search speed is now 280K nodes/sec in middle-game on a DEC Alpha/500.

Recent Improvements

- Mar 1995: 10-disc patterns
- May 1995: new best-first style endgame search for solving close positions faster
- Jul 1995: book-algorithm is now playing for maximal score when leaving the book
- May 1996: book-randomization
- Oct 1996: multi-ProbCut
- Nov 1996: major table estimation bug fixed, new version beats old 57% of the time
- Dec 1996: hash-table improvement saves 15-40% searchtime
- Mar 1997: a new table estimation technique increases the playing strength considerably
- Apr 1997: the entire book has been re-computed using the new evaluation function
- May 1997: speed boost on a DEC Alpha/500: in middle-game 280k nodes/sec, in endgame 1000k nodes/sec
- July 1997: increasing the number of training examples by means of book correction and generating rare positions to fill table gaps improves the evaluation function

---

Tournament Results (last updated: January/28/98)

Murakami vs. Logistello &
Workshop on Game-Tree Search
(last update: August/20/97 Match Report!)

Princeton I: Logistello wins 22 games in a row

---

last modified on *02/02/2011 22:33:43* ; you are visitor # 

since Dec/18/1995
to homepage of LOG's creator