Evolution of an Efficient Search Algorithm for the Mate-In-N Problem in Chess

Ami Hauptman and Moshe Sipper

Ben-Gurion University, Israel

2007 "HUMIES" AWARDS FOR HUMAN-COMPETITIVE RESULTS

Monday, July 9, 2007

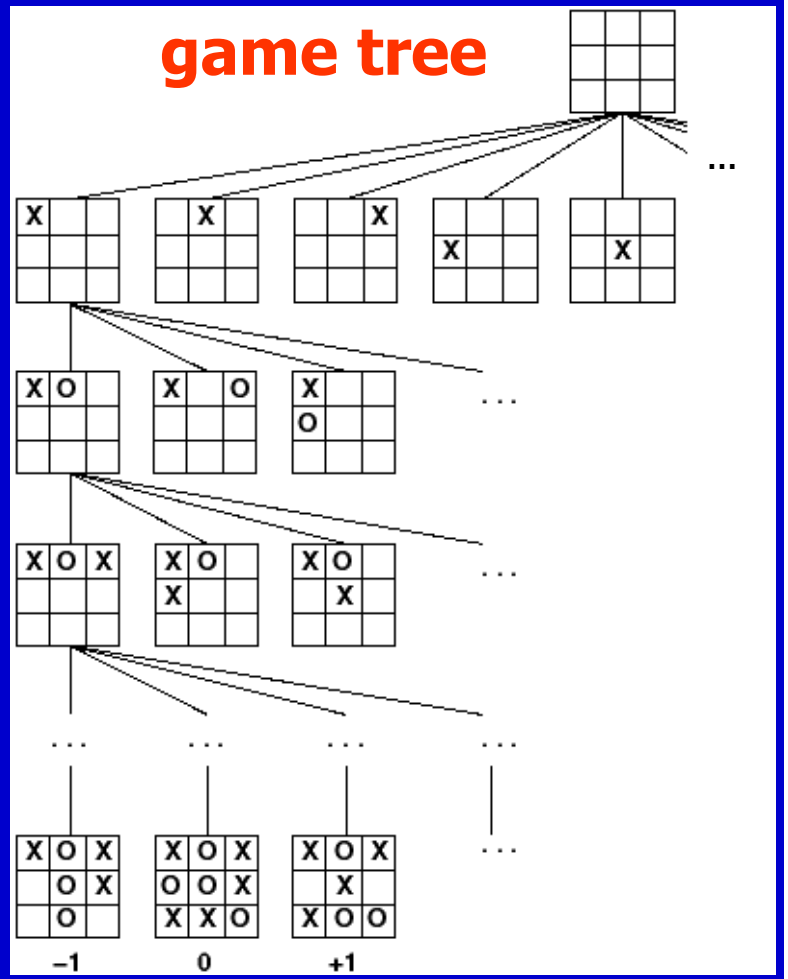
Game-Playing AI

- Game Strategy =

  Search + Knowledge
- Search:

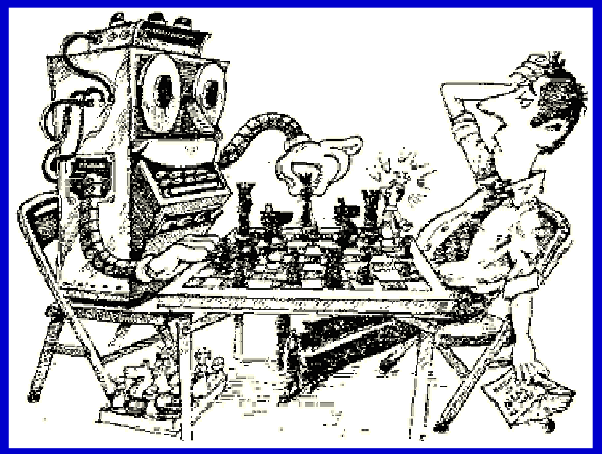
  Number of nodes developed
- Knowledge:

  Evaluation of nodes
- Tradeoff between the two



Chess: Machine Players

- Powerful contemporary engines
  - Crafty, Fritz, Deep Junior, …
  - Lots of search
  - Less knowledge
- Intelligent? Hmmm…
  - Very little generalization
  - Gobbles computational power
  - Deemed theoretically uninteresting [Chomsky, 93]



Chess: Human Players

- Use problem solving cognition
- Deeply knowledge-based play
- Massive use of pattern recognition; parallelism
- Also use search but
  - Substantially less nodes (typically dozens)
  - Selective (only "good")
  - More efficient: less nodes for "same" result
- Good source of inspiration for algorithms

Our Goal

- Concentrating on endgames we previously:
  - evolved node-evaluation function (knowledge) with GP
  - Results: draw or win against CRAFTY, a world-class chess engine
  - Part of work that won a 2005 humies medal
- This work: Evolve the search algorithm itself
- Evolve both search and knowledge, letting evolution balance the two

Incentive for Current Work

- Previously evolved players:
  - Sometimes miss (easy) shallow mates
  - Scaling problem: adding pieces to board decreased scores
- Evolved players should rely more on search
  - Full pure-knowledge player still unattainable
  - Search makes the strongest engines
- Problem:
  - Simply adding search: too slow (each node thoroughly examined)

➔ SOLUTION:

  - Balancing search & knowledge through **evolution**

Problem Domain

- Mate-in-N: Is there a forced win sequence in maximum 2*(N-1) plies ?
- Crucial to chess engines, searched far more thoroughly
- CRAFTY: For difficult N=5 cases searches over 10^6 nodes

Mate-in	1	2	3	4	5
Depth in plies	2	4	6	8	10
Nodes developed	600	7K	50K	138K	1.6M

Major Result

Evolved search algorithm:

Number of nodes developed reduced by 47%

with respect to world-class engine (not simple αβ)

Mate-in	1	2	3	4	5
CRAFTY	600	7K	50K	138K	1.6M
Evolved	600	2k	28k	55K	850k

Result is Human-Competitive

(H) result holds its own or wins a regulated competition involving human-written computer programs

(B) better than result accepted as a new scientific result at the time

(D) result is publishable in its own right

(F) better than result considered an achievement at the time

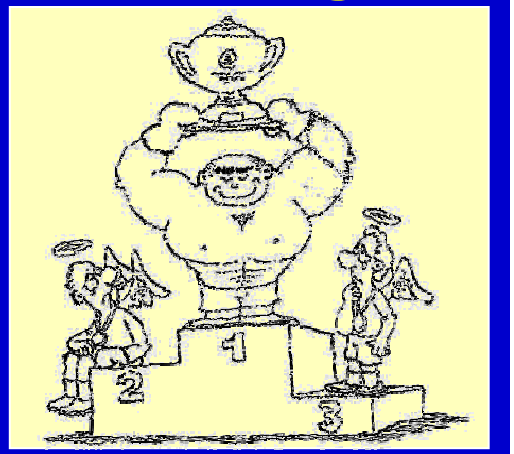
(G) result solves a problem of indisputable difficulty in its field

Why is Result Best?

- Difficult for most human chess players:
  - Must train intensively not to miss (and lose game)
- Our evolved strategies improve upon one of top chess engines in existence (Crafty), representing many human years of programming
- We're beating this top-notch engine in its own "territory": massive search
- Problem is crucial to chess engines, therefore much computational power is expended (e.g., in such positions, Deep Blue examines twice the normal number of nodes)

Why is Result Best? (cont'd)

- Evolving a dynamic algorithm (i.e., a process) usually harder than evolving a static structure
- We took evolution to the next level: balancing search and knowledge
- Surpasses previous EC solutions



In a nutshell:

1. Hard problem in hard domain for man & machine (chess)
2. Evolved algorithm better than (most) humans
3. Evolved algorithm better than human-written top engine
4. Evolution taken to next level

- A. Hauptman and M. Sipper

  Evolution of an efficient search algorithm for the mate-in-n problem in chess

  Proceedings EuroGP2007, pp. 78-89, April 2007

- M. Sipper, Y. Azaria, A. Hauptman, & Y. Shichel

  Designing an evolutionary strategizing machine for game playing and beyond

  IEEE Transactions on Systems, Man, and Cybernetics, Part C, 37(4), July 2007