

CORRESPONDENCE

SOME COMMENTS ON PROBLEM-SOLVING ABILITY TESTS

*Ilkka Blom*Salmikatu 27 A 16
65200 Vaasa 20, Finland

With great interest I read the articles in the ICCA Journal (Vol. 8, Nos. 2-4) on the problem-solving ability of various chess-playing computers and problem-solving programs for home computers. In the discussion contained in these articles, my problem-solving programs, Matebadix and Alybadix, were referred to.

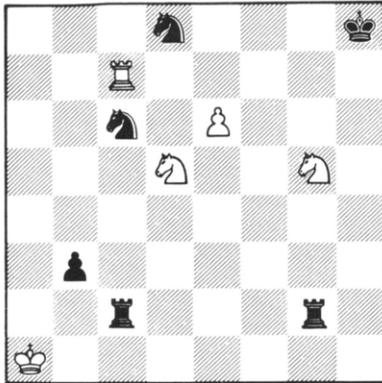
The present version of the Alybadix solving programs for Apple II computers I developed completely independently of other programs and solving theories. I worked on them mainly in the years 1980 to 1984. In 1984-85 I converted them for the Commodore-64. It was interesting to read in Mr. Wiereyn's article 'Inventive Problem Solving' (Vol. 8, No. 4, pp. 230-234) that we both have found many similar tricks for speeding up problem solving.

However, there is one principal shortcoming in the comparison of the solving times; all the times of Matebadix Intelligent were *test* times including testing *all* solutions. I think that it is not very reasonable to compare only the times it takes to find *one* solution since *chance* may then strongly affect the time (e.g. in Matebadix Intelligent the order in which the pieces are set up matters)! Moreover, in chess-problem testing, it is extremely important to find *all* possible solutions, since the problem is not correct if there are unintentional additional solutions (so-called cooks). Thus Mr. Wiereyn's comparison table (*ibid.*, p. 234) should be as follows:

No.	Wiereyn 3.3 MHz new (+)	Matebadix Intelligent 3.3 MHz
1	0.23	1.2
2	0.43	2.1
3	0.28	1.8
4	9.36	16.4
5	4.43	9.7
6	3.25	11.2
7	6.88	23.6
8	98.7	194.2
9	147.1	237.3
10	134.3	229.1
11	62.6	134.8
12	192.2	245.5
13	284.2	97.6
14	16.8	21.8
15	1037.5	207.6
16	792.8	1014.2
T:	2791.1	2448.1

If Apple II with 3.5 MHz accelerator card is used, the Matebadix *real* testing times would be a little better than in this table, where the times have been converted uniformly to 3.3 MHz from Commodore-64 (1 MHz) times.

One good trick to speed up the solving is to program Black to find strong defensive moves such as checks and captures. For instance, the following problem was tested completely by Matebadix Intelligent (on Commodore-64) in about 11 seconds.



Mansuba
Dr. L. Lindner's test

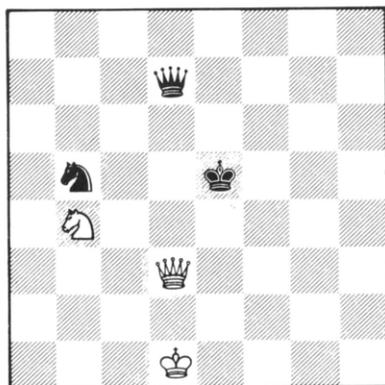
White: Ka1 Rc7 Nd5 Ng5 Pe6 (5)
Black: Kh8 Rc2 Rg2 Nc6 Nd8 Pb3 (6)
Mate in five.

Computer solution: 1. Rc7-h7+ Kh8-g8 2. Nd5-f6+ Kg8-f8 3. e6-e7+ Nc6xe7
4. Rh7-f7+ Nd8xf7 5. Ng5-e6 mate.

Number of moves: 2,811 (White: 1,572; Black: 1,239)

The most challenging problem genre for me has been the *helpmate*, in which Black helps White to mate Black with Black starting. Solving more-mover helpmates with a computer is generally very time-consuming. Helpmates are suitable for comparing the speed of different solving programs, since the possibility of chance affecting solving times is slight.

The Alybadix series includes three programs for helpmates: Helpbadix, Helpbadix C and Helpbadix Intelligent. One example of Alybadix' performance is presented below.



M. Luchin (U.S.S.R)
Problem Observer (Nov., 1985)

White: Kd1 Qd3 Nb4 (3)
Black: Ke5 Qd7 Nb5 (3)
Helpmate in two (3 solutions).

Solutions:

1. Nb5-d6	Qd3-f3	2. Qd7-e6	Nb4-c6	mate.
1. Nb5-d4	Qd3-g6	2. Qd7-d5	Nb4-d3	mate.
1. Qd7-e6	Qd3-f3	2. Nb5-d6	Nb4-c6	mate.

An estimation of the number of different moves amounts to $25^4 = 390,625$. In *Problem Observer*, Mr Ling tested the problem with the good playing program Colossus 4.0 (by Martin Bryant) on Commodore 64. Alybadix' times, when running on an Apple II 1 MHz, are about 10 percent faster than when running on a Commodore 64.

The times and the number of moves are as follows:

Program	Testing Time	Number of Moves		
		All	White	Black
Colossus 4.0	16 min 14 sec	390 027		
Helpbadix	33 sec	15 485	2 210	13 275
Helpbadix C	4 sec	840	709	131
Helpbadix Intelligent	1 sec	314	243	71

Colossus can be seen to have tried all possible moves. The Alybadix programs can prune moves according to the following rules:

- Helpbadix can cause cutoffs by White's last (mating) moves only;
- Helpbadix C can cause cutoffs by Black's and White's last moves;
- Helpbadix Intelligent can cause cutoffs for all moves.

The solving method of Helpbadix Intelligent is completely different from that of the other programs. It performs best in miniature helpmates. Most 2-6 move miniature helpmates can be solved in a reasonable time with Helpbadix Intelligent. Helpbadix C is especially strong in many-piece problems, in particular if there are relatively many flying (i.e., unguarded) squares around the Black King.

To facilitate the development of faster problem-solving programs, all kinds of solving tests are welcome and useful for the programmers. I am sure that many tricks for speeding up problem solving may be applied to chess-playing programs, although I myself have not yet had time for this.

COMMENTING BLOM'S COMMENTS

Paul Wiereyn

M. Nijhoffstraat 202
1382 TW Weesp, The Netherlands

In my article (Vol. 8, No. 4, pp. 230-234) I based my comparison of solving times on G. Grottlings contribution in the ICCA Journal (Vol. 8, No. 2, pp. 107-110). In his contribution, the table (ibid., p.110) bears the following footnote: "The solution times have been given in seconds, counting to the instant when the computer declared its solution to be final." This implies that the times recorded are times up to the key move having been found.

To the best of my knowledge, some of the computers mentioned in the table do not (or even cannot) search further. Considering Blom's comment in this issue, Grottlings' table turns out not to be completely correct (obviously because of confusion of notions). In my opinion, a footnote should be inserted in Blom's comment accordingly.

I would like to add that Blom's multifaceted solving program is a great achievement. Like Blom, I never seem to have enough time to realize all my ideas. At the moment I am porting my program to an Atari 800 XL which is a big task since practically all registers, variables and memory addresses must be changed.