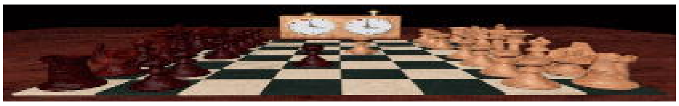


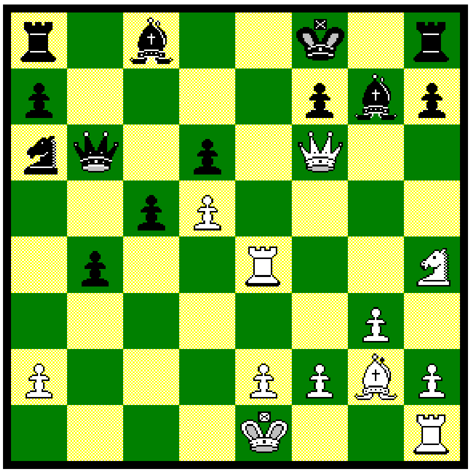
Chess Tests: Basic Suite, Positions 26-30

(c) Valentin Albillo, 2020

Last update: 14/01/98

See the *Notes on Problem Solving*

26.- Cukierman vs. Tartakover



FEN: r1b2k1r/p4pbp/nq1p1Q2/2pP4/1p2R2N/6P1/P3PPBP/4K2R/ w

White to play and mate in 9:
1. Qf6-e7+ Kf8-g8 2. Qe7-e8+ Bg7-f8

Results

Program	CPU/Mhz	Hash table	Move	Value	Plys/Max	Time	Notes
Chess Genius 1.0	P100	320 Kb	Qf6-e7+	Mate10	7/19	00:00:46	seen at 22s
Crafty 12.7	P100	12+5 Mb	Qf6-e7+	Mate9	8/17	00:00:13	
KAI Crafty 12.6	Pentium Pro 200 MHz	24+16 Mb	Qf6-e7+	Mate9	8/17	00:00:02	
KAI Chess Master 5500	Pentium Pro 200 Mhz	?	Qf6-e7+	Mate9	8	00:00:03	
KAI MChess Pro 5.0	Pentium Pro 200 Mhz	10 Mb	Qf6-e7+	Mate9	7	00:00:38	

Notes:

Cukierman announced mate in 10 to his adversary. I don't know how long he thought about it. Anyway, he was slightly wrong, as this position is really a Mate in 9, not 10.

Chess Genius 1.0 takes very little time for this mate, but it des not find the shortest mate in 9, but rather a mate in 10, probably the same that Cukierman announced. It needs just 7 full plies of search plus 12 extra plies for checks, thus adding to the 19 plies necessary to see a mate in 10.

Crafty 12.7 does pretty well for this kind of problems, specially when there are a little number of replies possible (most black moves are forced), and so finds the correct mate in 9, and does so 3 times faster than CG1.0

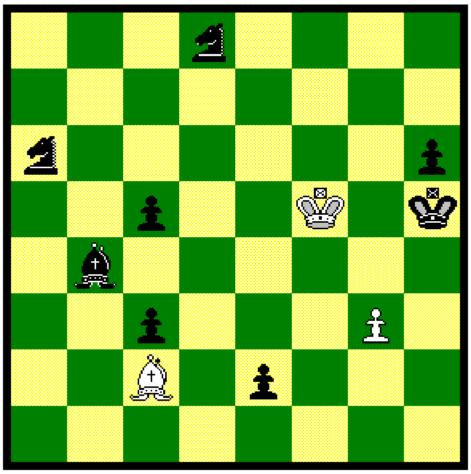
Crafty 12.6 does exactly the same as **Crafty 12.7**, but running in faster hardware and with much more RAM, it does it *six times faster* no less !.

Both **Chess Master 5500** and **MChess Pro 5.0** manage to also find this mate in 9 pretty fast. Notice, however, that while **Chess Master 5500** looks at 8 plies but achieves one of the shortest times, **MChess Pro 5.0** looks at one ply less (as usual) yet it takes more than *10 times longer*.

Addendum:

I stated originally that this problem was a mate in 10, because that is what Cukierman announced, and that was what CG1.0 found. But **Kai Luebke** sent an e-mail telling me that it really was a mate in 9, and I later confirmed this with **Crafty 12.7**. Thanks, Kai.

27.- Final part of a study of Gijs van Breukelen



FEN: 3n4/8/n6p/2p2K1k/1b6/2p3P1/2B1p3/8/ w

White to play and win: 1. Bc2-e4

Results

Program	CPU/Mhz	Hash table	Move	Value	Plys/Max	Time	Notes
Chess Genius 1.0	P100	320 Kb	Bc2-e4	Mate9	13/25	00:00:26	seen at 22s
NEW Comet-A.75	P100	13786 Kb	Bc2-e4	Mate9	16	00:14:36	24.986.369 nodes
NEW Rebel Decade 2.0	P100	512 Kb	Bc2-e4	Mate9	15	00:08:55	7.428.145 nodes
KAI Crafty 12.6	Pentium Pro 200 MHz	24+16 Mb	Bc2-e4	Mate9	17	00:00:46	
KAI Chess Master 5500	Pentium Pro 200 Mhz	?	Bc2-e4	-11.92	11	00:10:00	can't see mate

Notes:

In this *amazing study*, White wins with just a Bishop and a very backward, non-passed pawn against Black's two knights, bishop, and four pawns, one of them about to promote !

The key, of course, is Black's King *dangerous position* at the border of the board, and blocked by its own pawn. Nevertheless, it's incredible that White manages to not only avoid defeat but to give instead *a mate in 9*.

Due to the forced nature of the moves, and the use of hash tables, **Chess Genius 1.0** can search to 13 full plies (plus another 12 extra) in a very short time, discovering the 17-ply mate as part of the routine.

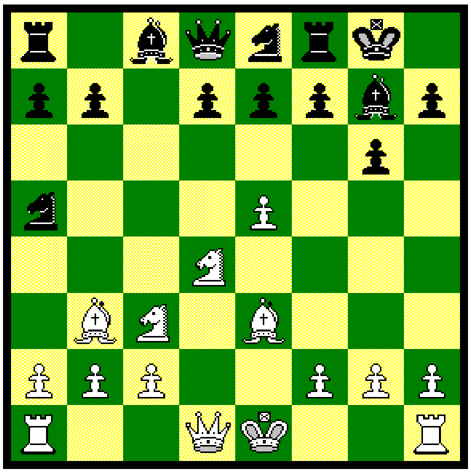
Comet-A.75, a freeware program, needs to look at 16 plies to find the mate in 9, but though it uses a large 13 Mb hash table, it has to evaluate nearly *25 million positions*, so it takes very long, more than *30 times longer* than CG1.0.

Rebel Decade 2.0, also freeware, finds the mate in 9 in one ply less, 15 plies, and despite its smallish 512 Kb hash table, it needs to evaluate only *7.428.145 positions*, so it's nearly *2 times faster* than **Comet-A.75**, but still *20 times slower* than CG1.0.

Crafty 12.6 also finds the winning move, and recognizes it as a mate in 9, but needs 17 plies to do so, instead of CG's 13 plies, and thus it does it almost *two times slower*, despite the much faster hardware and larger RAM. Still, it runs rings around both **Comet-A.75** and **Rebel Decade 2.0**.

Most surprisingly, **Chess Master 5500** *fails to find this mate* even when looking at 11 plies taking a long time, from *10 to 20 times longer* than the other programs tested. Which is worse, it evaluates the position as *very negative* for white, which actually can mate black in 9 !. If anyone knows or conjectures a reason for this, I would like to know ! Else, it seems either a bug or a very bad handling of this specific position.

28.- Fischer vs. Reshevski



FEN: r1bqnrk1/pp1pppbp/6p1/n3P3/3N4/1BN1B3/PPP2PPP/R2QK2R/ w

White to play and win the queen: 1. Bb3xf7+

Results

Program	CPU/Mhz	Hash table	Move	Value	Plys/Max	Time	Notes
Chess Genius 1.0	P100	320 Kb	Bb3xf7+	+3.03	5/17	00:00:07	wins the queen
KAI Crafty 12.6	Pentium Pro 200 MHz	24+16 Mb	Bb3xf7+	+3.194	7	00:00:02	wins the queen
KAI Chess Master 5500	Pentium Pro 200 Mhz	?	Bb3xf7+	+3.74	8	00:00:04	

Notes:

Fischer was a teenager when he gave this nasty surprise to poor old Sammy **Reshevski**. Just out of the opening, he *won the enemy Queen* with this spectacular move. The key is that the Queen is *blocked* by its own army, lacking lines to escape.

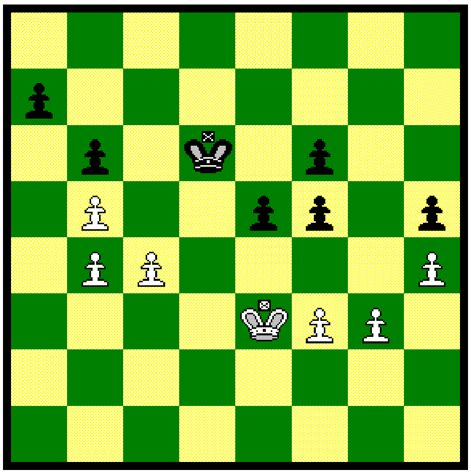
This is not a difficult position for a modern chess program and some of them even have it programmed as part of their openings book so it's found with no search at all. If your program plays it *instantaneously*, you know it was in the book.

Chess Genius 1.0 finds the winning move very quickly, in just 5 plies of full search, though the usual 12 extra plies help a lot, because there is a variation in which Blacks tries to avoid losing the Queen by moving instead his King in the open, but receives a forced mate. Once **Chess Genius 1.0** discovers that possibility, it sees that Black must lose its queen or else get mated.

Crafty 12.6 also finds the winning move, and thanks to its faster hardware, some *three times faster* than CG1.0, though the hardware-corrected times should be quite similar. Notice however that it has to search to 7 plies, while CG1.0 finds the winning move while searching to only 5 plies.

Finally, **Chess Master 5500** finds too the winning move, and though it looks at 8 plies, instead of 5/17 plies like **Chess Genius 1.0**, it does it in the same time approximately.

29.- Taken from "How computers play chess", pag. 132



FEN: 8/p7/1p1k1p2/1P2pp1p/1PP4P/4KPP1/8/8/ w

White to play and win: 1. g3-g4

Results

Program	CPU/Mhz	Hash table	Move	Value	Plys/Max	Time	Notes
Cray Blitz	Cray	Yes	g3-g4	+5.0	13	00:01:00	sees promotion
Chess Genius 1.0	P100	320 Kb	g3-g4	+3.81	7/19	00:00:25	selec=12
Chess Genius 1.0	P100	320 Kb	g3-g4	+3.81	10/10	00:00:07	selec=0
Rebel Decade 1.2	P100	192 Kb	g3-g4	+2.16	13	00:02:58	
NEW Rebel Decade 2.0	P100	512 Kb	g3-g4	+2.56	14	00:10:06	
Crafty 12.7	P100	12+1 Mb	g3-g4	+2.920	11/20	00:32:16	seen at 9m26s
Crafty 12.7	P100	12+1 Mb	g3-g4	+3.790	12/16	01:52:34	
Crafty 12.7	P100	12+1 Mb	g3-g4	+5.514	13/23	05:09:53	sees promotion
Crafty 12.7	P100	12+1 Mb	g3-g4	+6.160	14/21	11:32:22	sees promotion
HYATT Crafty 13.3	P6	?	g3-g4	?	11	00:04:10	see notes
KAI Crafty 12.6	Pentium Pro 200 MHz	24+16 Mb	g3-g4	+2.410	10	00:02:35	seen at 44s
KAI Chess Master 5500	Pentium Pro 200 Mhz	?	g3-g4	+3.08	11/19	00:02:50	seen at 1:06

Notes:

A many-pawn endgame, and the *only* move that secures the win is g3-g4.

Cray Blitz sees some 3 to 6 plies deeper than **Chess Genius 1.0**, and so sees more gain, in a very short time. **Chess Genius 1.0** takes also very little time to search 7 full plies plus 12 extra capture plies, and sees an almost 4-pawn gain.

Observe that if the number of extra plies is reduced from 12 to 0, the saving in time at the endnodes of the search allows for 3 more plies of full search, and even so the search proceeds almost *4 times faster* to reach exactly the same result.

Rebel Decade 1.2 needs 13 plies to see a substantial gain, though it doesn't take too long, looking at some *2.227.430 positions*. Notice that it does not see a 5 pawn gain as **Cray Blitz**, despite apparently similar depths, but it takes a fraction of **Crafty**'s time for a similar evaluation, even though its 192 Kb cannot compare to **Crafty**'s 12 Mb one.

Rebel Decade 2.0 does more or less the same as the older version. It looks one ply deeper, at 14 plies, and finds the correct move with a reasonable +2.56 evaluation. Going one ply deeper, at 15 plies and more than *19 million positions* examined, takes 22m 54s and raises the evaluation to +3.82, which compares badly with CG1.0, which needed just 7 plies in 25 seconds to get a +3.81 evaluation. On the other hand, the comparison against the various versions of **Crafty** is more balanced.

Crafty 12.7, using a very large 12 Mb hash table (40 times larger than that of **Chess Genius 1.0**) takes very long to find the correct move, but as it goes deeper and deeper, it keeps on finding more correct moves in the main variation, seeing larger and larger gains, from a few pawns to the *unavoidable promotion*. The time required almost *triples* when searching a ply deeper.

Crafty 12.6 searches to 10 plies and finds the correct move, but it does not see the promotion at this depth, and sees less gain than CG1.0, taking also more plies and more time, comparable to that of RD1.2.

Same problem for **Crafty 13.3**. The analysis sent by **Robert Hyatt** shows that it takes 11 plys and a somewhat long time on his very powerful hardware just to see that **g4** is the best move, see **Addendum** below for an explanation.

Chess Master 5500 sees the correct move, but not the promotion. It is worth noting that it searches to 11/19 plies, comparable to **Crafty**'s 11/20 plies, and finds the same gain (+3.08) as **Crafty** (+2.92) more or less, but does it nearly *6 times faster*, perhaps due to larger hash tables ?.

Addendum:

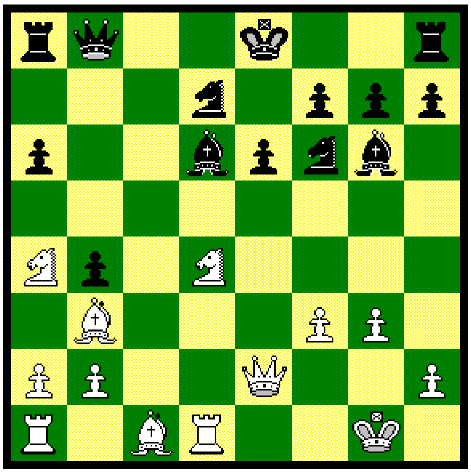
I sent an e-mail to **Robert Hyatt** with this comment:

VA: *" ... Here **Crafty 12.6** needs much more time and sees less gain looking at 10 plies than **Chess Genius 1.0** looking at just 7 !?"*

RH: *"Not uncommon. **Kd3** seems to be another way to play this, creating a passed pawn on the queenside. Takes **Crafty 13.3** 4 minutes to decide to play **g4**, which is just slightly lower in score than **Kd3** ... until it finally breaks over the top ..."*

```
depth   time   score   variation (1)
  9     4.28   1.471   Kd3 Ke7 c5 Ke6 c6 Kd6 Ke3 Ke7 Kf2 f4
  9->   6.53   1.471   Kd3 Ke7 c5 Ke6 c6 Kd6 Ke3 Ke7 Kf2 f4
 10    10.62   1.493   Kd3 Ke7 c5 Ke6 c6 Kd6 Ke3 Ke7 Kf2 e4
                       c7 Kd7
 10->   1:28   1.493   Kd3 Ke7 c5 Ke6 c6 Kd6 Ke3 Ke7 Kf2 e4
                       c7 Kd7
 11     1:40   1.315   Kd3 Ke7 c5 Ke6 c6 Kd6 Ke3 Ke7 Kf2 e4
                       fxe4 fxe4 Ke3
 11     4:10      ++   g4!!
```

30.- Chaos vs. Chess 4.0



FEN: rq2k2r/3n1ppp/p2bpnb1/8/Np1N4/1B3PP1/PP2Q2P/R1BR2K1/ w

White to play and win: 1. Nd4xe6

Results

Program	CPU/Mhz	Hash table	Move	Value	Plys/Max	Time	Notes
Chess Genius 1.0	P100	320 Kb	Nd4xe6	+0.33	3/15	00:00:04	seen at 00:00:01
Chess Genius 1.0	P100	320 Kb	Nd4xe6	+1.90	5/17	00:00:35	
KAI Crafty 12.6	Pentium Pro 200 MHz	24+16 Mb	Bxe6	+0.15	11	00:05:31	can't see it
HYATT Crafty 13.3	P6	?	Bxe6	-0.052	10/13	00:01:25	see notes
KAI Chess Master 5500	Pentium Pro 200 Mhz	?	Be3	-0.18	9	00:12:32	can't see it

Notes:

This is not a difficult position but *a historic one*. This was played in a game between two early chess programs, **Chaos** and many-times-champion **Chess 4.0**, in Stockholm, 1974

Chaos astonished everyone with this *Tal-like Knight sacrifice*, which was afterwards cited in all references to computer chess as an outstanding (and the first!) example of *a positional sacrifice*.

Now modern chess programs find this *positional* move almost instantly, and discover material gains very soon, with very, very modest ply depths by actual standards.

For instance, **Chess Genius 1.0** sees the move at once, in less than a second, and only needs a few seconds more to reach a depth of 3 plies, where the move appears strong enough to compensate for the loss of the knight and even give *some advantage*, and then, in very little time, it reaches 5 plies (plus 12 extra at the endnodes) which shows a net *gain* of nearly 2/3 of a Knight. Poor old **Chaos** probably did not see any of this !

Kai Luebke reports that **Crafty 12.6** does *not* find the knight's sacrifice, which seems to me very strange indeed ! It searches very deep, to 11 plies, takes a long time, yet it finds neither the correct move nor any substantial gain. Anyone can *explain* ?

Also, **Robert Hyatt** sent an analysis of **Crafty 13.3** for this position, and it also fails to find the knight sacrifice, even after looking at 10/13 plies. See the **Addendum** below for **Hyatt**'s comment and analysis. However, I'm not convinced by what he says, because the evaluation of his chosen move is -0.052, while **Chess Genius 1.0** evaluates the knight move as +1.90, substantially better. I don't believe that a drawish -0.052 evaluation can be considered as a sign of black's position wrecking.

Another surprise is **Chess Master 5500** which looks at 9 plies taking a very long time, yet *it does not find* the winning move, and the move selected is even evaluated a little *negatively for white* ! Does anyone know or conjecture why ?

Addendum:

I sent an e-mail to **Robert Hyatt** with this comment:

VA: *" ... Another one unfathomable to me.* ***Crafty 12.6*** *does not find this extremely old, much publisized, simple knight sacrifice even when looking at 11 plies requiring a long time, while other programs find this almost instantly, looking at as little as 3 plies !!??"*

RH: *"This is another animal. No one ever said that* ***Nxe6*** *wins here. Just that* ***Chaos*** *played it and won. I was at that tournament in fact. however, current* ***Crafty 13.3*** *plays* ***Bxe6*** *instead which also wrecks the black position pretty well..."*

```
depth   time   score   variation (1)
  5     0.24  -1.292   Bc4 Ne5 Bg5 Nxc4 Qxc4 Nd5
  5     0.37  -1.265   Bg5 Kf8 Nc6 Qc7 Rac1
  5     0.44      ++   Nc6!!
  5     0.50  -0.753   Nc6 Qc7 Bxe6 Kf8 Qd2 Qxc6 Bxd7
  5     0.72  -0.523   Bxe6 Kf8 Bc4 Nb6 Nxb6 Qxb6
  5->   0.80  -0.523   Bxe6 Kf8 Bc4 Nb6 Nxb6 Qxb6
  6     1.14  -0.428   Bxe6 Kf8 Nf5 fxe6 Nxd6 e5
  6->   1.35  -0.428   Bxe6 Kf8 Nf5 fxe6 Nxd6 e5
  7     2.42  -0.244   Bxe6 Kf8 Nf5 fxe6 Nxd6 e5 Be3
  7->   3.01  -0.244   Bxe6 Kf8 Nf5 fxe6 Nxd6 e5 Be3
  8    11.24  -0.026   Bxe6 Ne5 Bh3 Kf8 Bg2 Qc7 f4 Bh5
  8->  12.37  -0.026   Bxe6 Ne5 Bh3 Kf8 Bg2 Qc7 f4 Bh5
  9    23.00  -0.101   Bxe6 Kf8 Bxd7 Nxd7 Nc6 Qc7 Qd2 Qxc6
                       Qxd6+ Qxd6 Rxd6 Ne5
  9->  29.88  -0.101   Bxe6 Kf8 Bxd7 Nxd7 Nc6 Qc7 Qd2 Qxc6
                       Qxd6+ Qxd6 Rxd6 Ne5
 10    59.46  -0.052   Bxe6 Kf8 Bxd7 Nxd7 Nc6 Qc7 Qd2 Qxc6
                       Qxd6+ Qxd6 Rxd6 Ne5 f4
 10->   1:25  -0.052   Bxe6 Kf8 Bxd7 Nxd7 Nc6 Qc7 Qd2 Qxc6
                       Qxd6+ Qxd6 Rxd6 Ne5 f4
```