ICGA JOURNAL

INTERNATIONAL COMPUTER GAMES ASSOCIATION

Vol. 36
No. 4
December 2013

Published by
Tilburg center for Cognition and Communication (TiCC)
Tilburg University
Tilburg, The Netherlands

ISSN 1389-6911

SOME INFORMATION ABOUT THE INTERNATIONAL COMPUTER GAMES ASSOCIATION (ICGA)

The ICCA (International Computer Chess Association) was founded in 1977 and represents the Computer Chess World vis-à-vis Computer Science Organizations, such as ACM and IFIP, and also vis-à-vis the International Chess Federation (FIDE). In 2002 the name of the Association was changed into International Computer Games Association (ICGA), thus incorporating the International Computer Chess Association (ICCA). In the same way the ICGA also represents the Computer Games world vis-à-vis the various international games federations for games other than chess.

World-wide membership comprises individuals as well as university and industrial members.

The activities of the ICGA are:

(i) to publish a quarterly *ICGA Journal*;
(ii) to hold regular World Computer-Chess Championships, Computer Olympiads, and *Advances in Computer Games* conferences;
(iii) to strengthen ties and promote cooperation among computer-games researchers;
(iv) to introduce computer games to the games world;
(v) to support national computer-games organizations and computer-games tournament organizers.

The ICGA takes pride in listing the roster of the ICCA Past Presidents:

1977 - 1983	Benjamin Mittman
1983 - 1986	Monroe Newborn
1986 - 1992	David Levy
1992 - 1999	Tony Marsland
1999 - 2011	David Levy

The setting of the copy for this issue was performed at the Tilburg University. The benevolence of the University Authorities is recognized for sponsoring part of the secretarial staff. Contributors are encouraged to read the **Information for Contributors** on page 233.

TABLE OF CONTENTS

FROM TILBURG TO LEIDEN

The *ICGA Journal* is moving for the fourth time in its existence. This time we move from Tilburg to Leiden. Your Editor believes that this move provides a suitable opportunity to inform our younger readers of some of the history of the ICGA and its Journal. Before doing so, we would first like to express our gratitude to the Tilburg University authorities for their generous and fruitful cooperation over more than five years. They gave us the space and means to edit and produce the Journals and to organise the ICGA events in Tilburg in 2011. In summary, the ICGA thanks Tilburg University for all their support.

With respect to the history of the ICGA we take the World Computer-Chess Championship (WCCC) in Stockholm (1974) as the starting point of our community, although some would argue that the North American Computer Chess Championships, that saw their first event in 1970, should be mentioned as a stimulating predecessor. The second WCCC took place in Toronto in 1977. There Barend Swets (Delft, the Netherlands) launched the ICCA and Ben Mittman (Evanston, IL) became the first Editor-in-Chief of the *ICCA Newsletter* and the first *ICCA President*. In 1983, he transferred the editorial function to Jaap van den Herik (then in Delft, in the neighbourhood of Swets). The *ICCA Newsletter* became the *ICCA Journal* (the first benefit of the 1983 move). In 1987, the *ICCA Journal* moved to Maastricht, in 2008 to Tilburg, and now (2014) to Leiden. As you can see the original name was related to chess only, since ICCA means International Computer Chess Association. During the Computer Olympiad in Maastricht 2000 the Journal's name was changed into *ICGA Journal.* Two years later, at the Triennial Meeting again in Maastricht, the name of the Association followed the change and so the ICCA was redubbed ICGA, meaning International Computer Games Association.

The reason for the current movement is that your Editor has now taken up a position in the Faculty of Science at Leiden University with the task of helping to establish the Leiden Centre of Data Science. He is pleased to mention that his move is taking place together with Joke Hellemons and Aske Plaat. It means that from January 1, 2014 the ICGA Headquarters are located at LIACS Leiden, the Netherlands. As you all may know Leiden has a rich history and many Noble Prize winners in physics. It is the oldest University city in the Netherlands and also well known for its Faculty of Law. Obviously, we look forward to welcoming our ICGA colleagues in this ancient and beautiful city. Although the ICGA headquarters are moving, the principles and themes of the Journal transcend geography and evolve apace.

The visiting address is: Leiden Institute of Advanced Computer Science (LIACS), Snellius Building, Room 164, Niels Bohrweg 1, 2333 CA Leiden, the Netherlands.

Thus the present issue both looks backwards and forwards, which is reflected in its contents. We thank Karsten Müller and Guy Haworth for their confirmation of the old theories on the famous chess game Timman-Velimirović and for formulating new ones concerning the ending Rook versus Bishop. Moreover, we are happy to publish a breakthrough in Chinese Chess Endgame Knowledge Bases. The authors deal with a very difficult topic and manage well to show us the importance of multi-level inference. The four notes demonstrate in a fine way the current progress in our community. Their contribution ranges from solving a minichess variant, via a new approach for algorithms to decide almost on the spot, to two contributions by Guy Haworth. His Chess Endgame News is nowadays a regular and well-known contribution which reads as a story in many parts. Over the years, as new *Depth to Mate* results have come in, many have wondered precisely how deep chess is, thinking ahead to 8-man, 10-man, and 32-man chess. The latest 7-man figures from Moscow seem to confirm a clear trend and Guy Haworth's statistical analysis has given rise to a stimulating conjecture, dubbed "Haworth's Law" by Thomine Stolberg-Rohr WFM. It will no doubt give rise to other conjectures. Then, we have Dap Hartmann's contribution, a laudable review of the Ph.D. thesis by Abdallah Saffidine. We conclude the issue with a series of tournament reports that show us progress, progress, and progress.

All in all, we see that many moves are made in our Games Community. It is up to readers and authors to make the next moves in any games they wish. Please do not forget to inform us about the results of your research. They will be reviewed and published for the benefit of our community.

Jaap van den Herik

The credits of the photographs in this issue are to: Henk Stoop, Professor X. Xu, Professor H. Iida, and Professor T. Cazenave.

It is also possible to receive your membership copy in electronic form. Please, arrange the change of your subscription with our Editorial Manager Johanna Hellemons.

ICGA Journal readers who are interested in information on our publications are referred to our website. A complete list of all articles, notes, and literature reviews published in the *ICCA Journal* and the *ICGA Journal* is accessible on the Internet at http://www.icga.org

Since October 1, 2013 all payments to the ICGA should be made to the ING Bank, the Netherlands. IBAN code: NL34INGB0003988921 / Swift code: INGBNL2A. Our Deutsche Bank account has been terminated.

ROOK VERSUS BISHOP

K. Müller and G.McC. Haworth[1]

Hamburg, Germany and Reading, UK

ABSTRACT

The focus here is on the influence of the endgame KRPKBP on endgames featuring duels between rook and bishop. We take advantage of the range of endgame tablebases and tools now available to ratify and extend previous analyses of five examples, including the conclusion of the justly famous 1979 Rio Interzonal game, Timman-Velimirović. The tablebases show that they can help us understand the hidden depths of the chess endgame, that the path to the draw here is narrower than expected, that chess engines without tablebases still do not find all the wins, and that there are further surprises in store when more pawns are added.

1. INTRODUCTION

There has been a tablebase revolution in the endgame rook against bishop. In general the theory of chess endgames is fairly stable compared to that of chess openings. It is very seldom that the theoretical verdict of a major cornerstone position is overturned but the complete solution of all endgames with seven men or less has of course changed several verdicts. For example, Ken Thompson created a KBBKN endgame tablebase (EGT) in 1983 which proved that in general two bishops win against a knight when the 50-move draw-claim rule is not taken into account (Roycroft, 1983, 1988). Human theory had thought that endgame was drawn if the defender reached the Kling-Horwitz position. Later John Nunn (2005) pointed out that amazingly, KQP(g)P(h)KQ is usually drawn if the defending king is well placed although human theory had assumed that the two extra pawns would win.

While in those two cases the evaluation of a whole type of endgame was changed, here we illustrate the influence of the KRPKBP tablebase on endgames featuring rook against bishop. Humans and computer engines without tablebases have big problems in several important positions as the dominance duels between rook and bishop can be surprisingly deep, difficult and incomprehensible to the human eye. One of the cornerstones of human theory has even been broken by computer analysis using the EGTs.

The following nomenclature and notation is useful:

DTC ≡ the metric 'Depth to Conversion', i.e. to mate and/or change of force (and *dtc* ≡ a DTC depth),
DTM ≡ the metric 'Depth to Mate' (and *dtm* ≡ a DTM depth),
DTZ ≡ the metric 'Depth To Zeroing of the ply count' (and *dtz* ≡ a DTZ depth),
SC^-M^- ≡ a move-choice strategy minimising DTC then DTM (and similarly, SC^-, SC^+, SM^- etc.),
° ≡ only move available, ′′′′ ≡ only value-retaining move,
′′′≡ only value-retaining move (after ignoring moves to a position four plies earlier),
′′ ≡ only optimal move, given the defined move-subsetting strategy (defaulted to $SM^{+/-}$), and
′ ≡ equi-optimal move, given the defined strategy

Today's endgame tables provide a definitive benchmark of endgame play as well as an opportunity to see how remarkably well the top players tend to play the endgame. The analyses here have been confirmed by one or more of Nalimov's sub-7-man DTM EGTs (Bleicher, 2014a; ChessOK, 2014a), FREEZER (Bleicher, 2014b; Rusz, 2014), Konoval's 6-man DTC EGTs (Konoval, 2014), the Lomonosov team's 7-man DTM EGTs (ChessOK, 2014b; MVL, 2014) and Romero's FINALGEN (2012).

These and other analyses may be played through and studied further using the accompanying pgn file and FREEZER EGTS available from Müller and Haworth (2014).

[1] HSK1830@aol.com. University of Reading, UK RG6 6AH. email: guy.haworth@bnc.oxon.org

2. SACHDEV-SCHUT (2012)

The first example here is a relatively easy 'warm up', a pure dominance duel in the 2012 game, Sachdev-Schut[2] (Chessgames.com, 2014a) starting with Figure 1a's position 56w. **56. Rc7**!? Nunn (2002) is a good reference here. White tries the best trick against the standard defence when Black's king is in the corner not controlled by the bishop. **56. … Be6?** Black falls for it. Among the drawing moves are 56. ... Ba2/Bd3=. **57. Kg6**! **Kh8**?! (This makes it relatively easy for White. 57. ... Bh3!? is the best try when White has only one way to win: 58. Re7!! (*58. Rf7? Bg2 59. Re7 Bc6 60. Re6 Ba4=*) 58. ... Kf8 59. Re5, Figure 1b. The central rook dominates the bishop. 59. … Bg2 60. Kf6 Bf3 61. Rf5 The rook forces the bishop to leave the shadow of the kings. 61. … Bc6 62. Rc5 Bd7 63. Rh5 Kg8 64. Rg5+ Kf8 (*64. ... Kh7 65. Rg7++-; 64. ... Kh8 65. Kf7+-*) 65. Rg1 Bc8 66. Rc1 Bd7 67. Rb1 Ke8 68. Rb8+ Bc8 69. Rxc8++-) **58. Rh7+! Kg8 59. Re7 1-0**, Figure 1c.

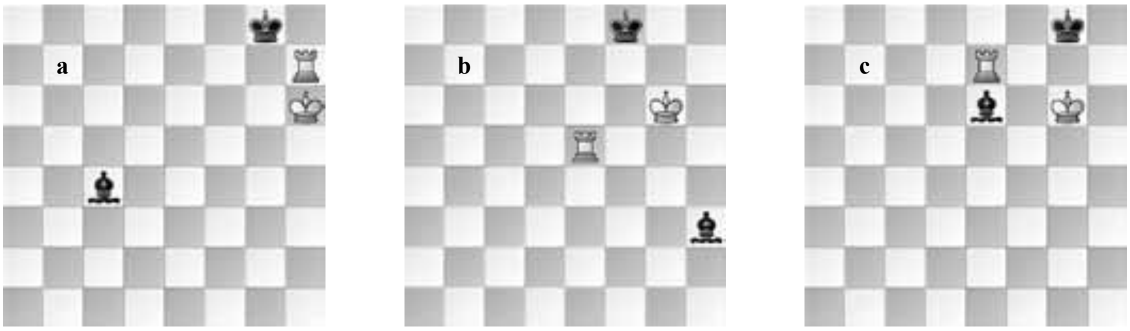


Figure 1. Sachdev-Schut (a) before 56w, (b) after sideline 59. Re5 and (c) after 59. Re7.

3. TIMMAN-VELIMIROVIĆ (1979)

The next example comes from the celebrated 1979 Rio Interzonal game Timman-Velimirović (Chessgames.com, 2014b), well known for the first appearance at the board of the KRP(a2)KBP(a3) endgame and for Timman's re-markable pre-emption of the expected 50-move draw-claim. It is also justly famous because of the initial analysis in 1948 by Chéron (1969) and the subsequent analysis by van den Herik and colleagues (1987, 1988a/b; Sattler, 1988), Timman himself (1981, 1996, 2011), Nunn (1981), and Müller and Lamprecht (2001).

As Timman (2011) says, Dvoretsky (2003) thought White should always win this endgame, and Chéron's work implicitly suggests as much. However, as Nalimov's DTM EGTs and Bleicher's FREEZER show, the game was drawn from KRPKBP position 64b until Velimirović's erroneous **68. … Kf8??** FREEZER finds 81% of wtm KRP(a2)KBP(a3) positions won but only 39% of btm positions lost.[3] Timman (1981) correctly outlined the safe zones for the Black king showing that Chéron's target positions could not always be reached.

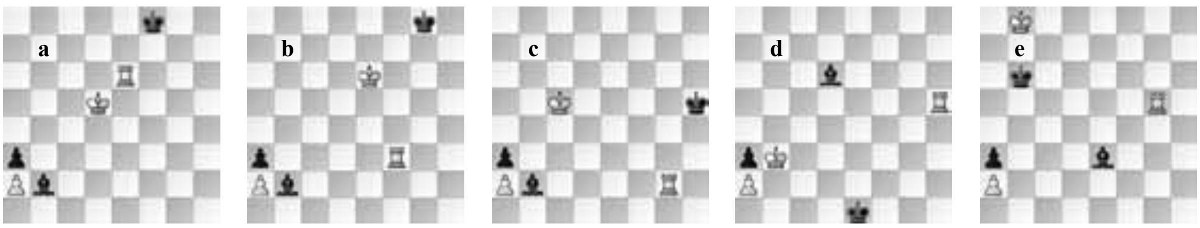


Figure 2. Timman-Velimirović: (a) main line 69w and (b) 78w, (c) after Line B's 81. Kc5, (d) after Line D, 100. Rh5. Off the board, (e) the maxDTC/Z KRP(a2)KBP(a3) position: *dtc/m* = 55/82m.

At the board, Timman had to contend with the FIDE draw-claim rule (of no interest to study enthusiasts including himself) but he was helped by his second, Ulf Andersson, during adjournments (Donner, 2007) at positions 44b, 64b and 78w. The goal is clearly to zero the ply-count before move 114b by mate, or by capture of the pawn or bishop: therefore the key metric is DTC. FREEZER and Konoval confirm that at 69w, *dtc* = 36 moves with best play but finding the win in time was a major challenge. In fact, Andersson and Timman improved on Chéron's

[2] Varying from FIDE's listing of her name, we recognise 'Tania' as Ms Sachdev's given name.
[3] The equivalent KRKB statistics are: 35% of wtm positions are won and only 3% of btm positions are lost.

"indispensable" analysis and found enough to achieve a confident and impressive win. Velimirović also had the benefit of Chéron's extensive analysis and put up a robust defence. Nevertheless, he never came close to the possibility of a 50-move draw-claim. As the following game line shows, annotated from FREEZER results relative to the DTC metric, neither player conceded more than 9 moves of depth in the next 35 moves:

Line A, 8/8/4k3/2p2r2/7b/p2K5/P7/5R2 w - - 1 64, game, =:
64. Rxc5 {KRPKBP, =: adjournment 2} **Bf6′′′′ 65. Rc6+ Ke7 66. Ke4 Bb2 67. Kd5 Kf7 68. Re6 Kf8??** {not …Kg7?? as in many sources. Figure 2a, 1-0, *dtc/m* = 36/56m. Ba1/c3/d4/f6/g7/h8 draw} **69. Ke4′′′′ Kf7′ 70. Kf5′′′′ Kf8″ 71. Kg6** {+1m} **Bc3** {-1m} **72. Ra6″ Bb2″ 73. Ra7″ Ke8″ 74. Kf5″ Kf8″** {*dtc/m* = 30/50m} **75. Ke6″ Kg8″ 76. Rf7″ Bc3′ 77. Rf3** {+1m} **Bb2″** {Figure 2b, (Chéron, 1969, p323; Timman, 1996, p26), *dtc/m* = 28/47m: adjournment 3} **78. Ke7″ Kh7′ 79. Rg3″ Kh6″ 80. Kd6!′ Kh5″ 81. Kc5′ Kh4″ 82. Rg8′ Be5″ 83. Kd5** {+1m} **Bb2′ 84. Kc4″ Bf6** {-2m, *dtc/m* = 20/39m} **85. Rg6″ Bg5″ 86. Kd3** {+2m} **Bc1** {-2m} **87. Ke4″ Bb2** {-1m} **88. Kf5** {+1m} **Kh5″ 89. Rd6** {+1m} **Kh4″ 90. Rd3″ Bc1″ 91. Rc3** {+1m} **Bb2″ 92. Re3!′ Bc1″ 93. Re1″ Bd2″ 94. Rh1+** {+1m} **Kg3° 95. Rd1″ Bb4″ 96. Rd3+′ Kf2′ 97. Ke4′ Ke2″** {*dtc/m* = 10/30m} **98. Kd4″ Bc5+′ 99. Kc4′ Be7′ 100. Rh3″ Bd6′ 101. Kb3′ Bf8** {-1m} **102. Rh8″ Bd6** {-1m} **103. Ra8″** {*dtc/m* = -2/-23m: 103. ... Kd2/3 inviting Rxa3?? but 104. Rd8″} **Resigns** 1-0.

The first computation of a 6-man EGT addressed this KRP(a2)KBP(a3) endgame (van den Herik, 1987) and provided the DTC-minimaxing line B below, confirmed correct by FREEZER:[4]

Line B, 5k2/8/4R3/3K4/8/p7/Pb6/8 w - - 9 69, SC$^-$/SC$^+$, *dtc/m* = 36/56m:
69. Ke4′′′′ Kf7′ {Kg7′} 70. Kf5′′′′ Kf8″ 71. Re4′ {Re1/2/3′} Kf7″ 72. Re3″ Bc1″ 73. Rc3″ Bb2″ 74. Rc7+‴ {and here, SM$^+$ prefers Kf8/g8} Ke8′ {Kf8/g8′} 75. Ke6″ Kf8″ 76. Rf7+″ Kg8″ 77. Ke7″ Kh8″ 78. Rf2′ {Rf1′} Kg7′ {Kh7′} 79. Rg2+″ Kh6″ 80. Kd6′ {Ke6′} Kh5″ 81. Kc5′ {Kd5′} {and here, Figure 2c, the bishop steps away from the pawn, q.v., line C below} Be5″ 82. Kb4′ {Kc4′} Bd6+″ 83. Kb3″ Kh4″ 84. Rg6′ {Rd2′} Be7″ 85. Kc3′ {Kc4′} Kh5″ 86. Rg2″ Kh4′ {Bd6′} 87. Kd3′ {Kd4′} Bf6″ 88. Ke4″ Bc3′ {Ba1/b2′} 89. Kf5″ Kh3″ 90. Rg4″ Be5″ 91. Kg5″ Kh2″ 92. Kg6″ Kh3′ {Bd6′} 93. Kf5″ Bd6″ 94. Ra4″ Kg2′ {Kg3′, Be7/f8′} 95. Ke4′ {Ke6′} Bf8″ 96. Kd3′ {Kd4/5′} Kf3″ 97. Kc3″ Ke3″ 98. Ra8″ Bd6″ 99. Ra6″ Bc5″ 100. Kc4″ Bf8″ 101. Ra8″ Bd6″ 102. Rd8″ Be5′ {Bc7/e7/f4/g3/h2′} 103. Rd3+′′′′ Ke4′ 104. Rxa3′′′′ {KRPKB, *dtc/m/z* = -8/-19/-2m} 1-0.

Perhaps at 68b in the game, Velimirović wished to continue the direct defence of his pawn. But the above line shows the bishop multitasking, exercising more control of the board, particularly of squares d4 and e5. The defence is foreshortened by 15 moves merely by constraining the bishop not to play 81. … Be5″ in Line B:

Line C, 8/8/8/2K4k/8/p7/Pb4R1/8 b - - 34 81, Figure 2c, SC$^-$/'constrained SC$^+$', *dtc/m* = -21/-42m:
81. ... Kh4 82. Kb4″ Kh3″ 83. Rg8′ Kh4″ 84. Kb3′ Kh3″ 85. Rg5′ Bc1″ 86. Rc5″ Bb2″ 87. Rc4″ Kg3′ 88. Ra4″ Kf3′ 89. Rxa3″ {KRPKB, *dtc/m/z* = -7/-19/-2m} 1-0.

Line B diverged unnecessarily from an SM$^+$ strategy at position 74b. The following minimaxes both DTC and DTM for a further 26 moves until position 100b, highlighting why these goals can conflict with each other:

Line D, 8/2R2k2/8/5K2/8/p7/Pb6/8 b - - 20 74, SC$^-$M$^-$/SC$^+$M$^+$, *dtc/m* = -30/-50m:
74. ... Kf8′ 75. Ke6″ Kg8″ 76. Rf7″ Bg7″ 77. Ke7″ Kh7″ 78. Rf2″ Bd4′ 79. Rg2″ Kh6″ 80. Ke6′ Bb2′ 81. Kd5″ Kh5″ 82. Kc4″ Be5″ 83. Kb3″ Bd6″ 84. Rd2″ Bf8″ 85. Rd3″ Kg5′ 86. Kc4″ Kf4″ 87. Kd5″ Be7″ 88. Ke6′ Bc5″ 89. Rc3″ Bf8″ 90. Rh3″ Ke4″ 91. Rh8″ Bc5″ 92. Rh4+″ Kd3″ 93. Kd5″ Be3″ 94. Rh3″ Ke2″ 95. Ke4″ Bc5″ 96. Rh2+″ Kf1″ 97. Kd3′ Bf8′ 98. Kc2′ Ke1″ 99. Kb3′ Bd6″ 100. Rh5′ {Figure 2d, *dtc/m* = -4/-24m. Black must lose the pawn earlier or hasten mate by losing the bishop later.} Bf8 *(SM$^-$C$^-$/M$^+$C$^+$: 100. ... Ke2′ 101. Ra5″ Ke3″ 102. Rxa3″ dtc/m/z = -10/-22/-2m)* 101. Rf5″ Bd6″ 102. Ra5″ Kd2″ 103. Rd5+″ *(100. Rxa3?? Bxa3′′′′ 101. Kxa3 Kc3′′′′=)* Ke3′ 104. Rxd6″ {*dtc/m/z* = -1/-12/-1m} 1-0.

The appendix and accompanying pgn file provide the maxDTC KRP(a2)KBP(a3) position (Figure 2e), the maxDTC and maxDTM KRPKBP positions, and appropriate depth-minimaxing lines from them.

[4] The EGT itself did in fact prove to have a few errors related to rare, unlikely and unconsidered positions (van den Herik et al, 1988b; Sattler, 1988; Timman, 1996, p143) but these were irrelevant to this game, the authors' sole focus.

4. ELKIES (1993)

In the next example, #4 of van der Heijden (2010) and Figure 3a, the computer was needed to break the defence. Human theory had thought that Black can draw but this is not the case as the rook can win the domination duel, a fact established by Noam Elkies in 1993. **1. Rb3 Bd6 2. Kg4** and White has three plans. He can invade with the king to f6 or h6 or play g5-g6 under the right circumstances. Black cannot frustrate all three plans. **2. … Bf8** (2. ... Bc5 3. Rb5 Bd4 (*3. ... Ba3 4. Kf5 Be7 5. Rb8+ Kf7 6. Rb7+- and White wins by bringing the king to h6.*) 4. Kh5 Bc3 (*4. ... Bg7 5. g6 h6 6. Rb8+ Bf8 7. Rxf8+ Kxf8 8. Kxh6+-*) 5. Rb8+ Kg7 6. Rb7+ Kh8 7. g6 h6 8. Kxh6 Bg7+ 9. Kg5 Bd4 10. g7+ Kh7 11. Rf7 Be5 12. g8=Q+ Kxg8 13. Kg6+-) **3. Kf5 Bc5 4. Rd3 Bb4 5. Kf6 Ba5 6. Rb3 Bd8+ 7. Kf5 Ba5 8. Kg4 Bc7 9. Rb5 Bd6 10. Kf5 Bc7 11. Rd5**, Figure 3b. The central rook dominates the bishop: **11. … Bb6 12. Kf6 Bc7 13. Rd7 Ba5 14. Rg7+ Kh8 15. Kf7+-**, Figure 3c, **1-0.**

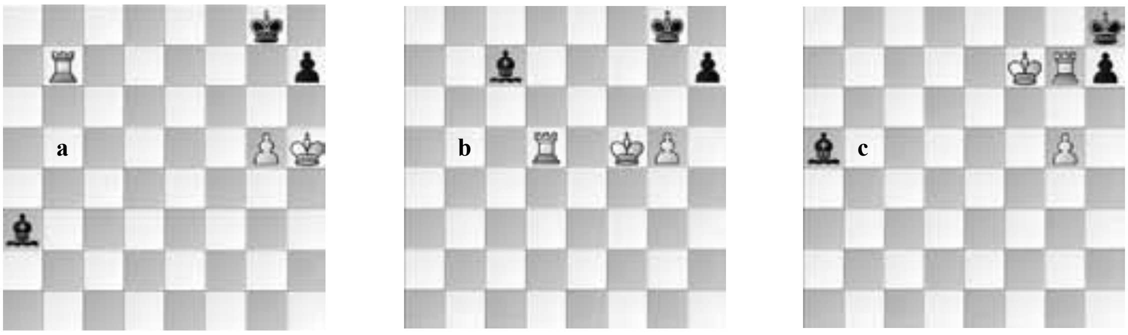


Figure 3. Elkies' study: main line positions (a) 1w, (b) 11b and (c) 15b.

5. GELFAND-IVANCHUK (2011)

The discussion of the next two positions is a slightly expanded version of Endgame Corner 143 (Müller, 2011). Position 54w from Gelfand-Ivanchuk (Chessgames.com, 2014) is of very high practical importance. Chess engines with 6-man EGTs could not find a win and it took FINALGEN, with the computational advantage of the facing pawns, to declare the position a fortress draw. But the drawing margin is not as large as it seems: Black must defend very carefully. It is not enough just to keep the bishop on the long diagonal and wait.

Figure 4a: **54. Rc2** (54. h4 Ba1= (*54. ... Bd4?, Figure 4b, is a typical mistake which often occurs in practical play. 55. Rc4 Bb2 56. g4 hxg4 57. Rxg4″″ Kh7 58. Kf7″″ Kh6 59. Rxg6+″″, KRPKB, Kh5 60. Rg2″″, dtc/m = -36/-50m, Figure 4c, and White eventually wins the domination duel, e.g., 60. ... Bc3″ 61. Rh2″″ Be1 62. Kf6 Bg3″ 63. Rh1″ Bf2 64. Kf5″ Be3″ 65. Rh2 Bg1 66. Rg2″ Bb6″ 67. Rb2″ Bc5 68. Rc2″ Be3″ 69. Ke4″ Bh6″ 70. Rh2″ Bg7″ 71. Kf4″ Bf8″ 72. Kf3″ Ba3″ 73. Ra2″ Be7″ 74. Re2″, Figure 4d, Bf6″* (74. ... Bxh4 is met by 75. Rh2 Kg5 76. Rh1+-, Figure 4e, very beautiful!). *75. Kf4″ Bd8″ 76. Rc2″″ Be7″ 77. Rd2″ Bb4″ 78. Rd8″ Bc3″ 79. Kg3″+-*)). (54. h3 Ba1 55. g4 hxg4″″ 56. hxg4 Bb2 57. g5 Ba1 58. Rf7 Bb2 59. Rf6, Figure 5a, just met by the calm Kg7″″=).

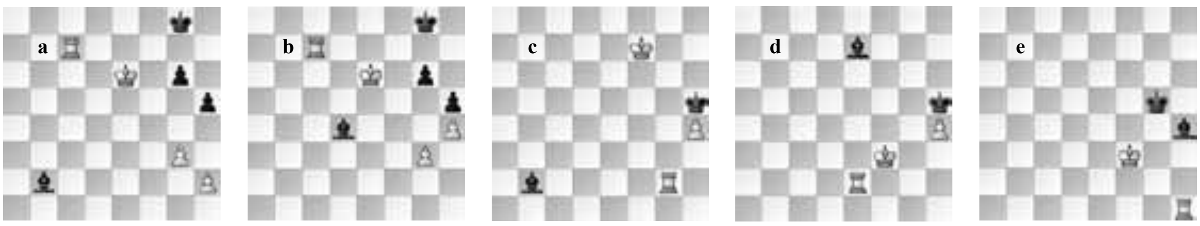


Figure 4. Gelfand-Ivanchuk: (a) 54w, and after (b) 54. … Bd4?, (c) 60. Rg2, (d) 74. Re2, and (e) 76. Rh1.

54. … Ba1 55. Rg2 Kg7 56. g4 hxg4″″ 57. Rxg4 KRPKBP **Bc3 58. Rc4 Ba1 59. Rf4 Bb2 60. Rf1 Bd4 61. Rf7+ Kg8 62. Rf4 Bc3 63. Rg4 Kg7″″** (63. … Kh7? 64. Kf7+-) **64. Rg2 Bf6 65. Rc2 Ba1 66. Rc7+ Kg8 67. h4 Bb2 68. Rc2!?** Figure 5b **Bd4″″** the only move. Black must indeed be very careful when defending this fortress. **69. Rd2** (69. Rg2 Kh7″″ 70. Kf7 Kh6″″ 71. Rxg6+ Kh5″″ 72. Rc6 Bf2″″ 73. Kf6 Kxh4″″=). After 69. Rc4, Figure 5c, the only move is the amazing Be3!! with the point 70. Kf6 Kh7″″ 71. Rg4 Kh6″″ 72. Rxg6+ Kh5″″ 73. Rg3 Bb6″″ 74. Rh3 Kg4 75. Rh1 Bd8+=. **69. ... Bc3 70. Rd3,** Figure 5d**, Be1″″**. Again Ivanchuk finds the only defence. The bishop must leave the long diagonal as 70. ... Bb2? runs into 71. Rg3 Kh7 72. Kf7 Kh6 73. Rxg6+ Kh5 74. Rg2 and White wins as seen in the line 54.h4 Bd4?

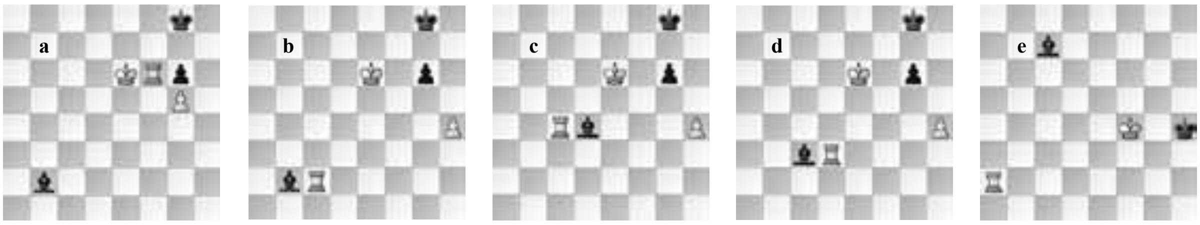


Figure 5. Gelfand-Ivanchuk after (a) 59. … Rf6, (b) 68. Rc2, (c) 69. Rc4, (d) 70. Rd3 and (e) 89. Bc7.

71. Kf6 (71. Rd4 Kg7 72. Rg4 Kh6 73. Kf6 Kh5 74. Rxg6 Kxh4 75. Kf5 Bd2=) **71. ... Bxh4+″″ 72. Kxg6** KRKB **Kf8″″ 73. Rh3 Bd8″″ 74. Rh7 Ke8 75. Kf5 Kf8 76. Ke6 Bg5 77. Rf7+ Kg8″″ 78. Rd7 Kf8 79. Rd5 Bc1 80. Rd1 Bb2 81. Rf1+ Kg7 82. Rf7+ Kg6 83. Rf2 Bc1 84. Rg2+ Kh5 85. Kf5 Kh4″″ 86. Rc2 Be3″″ 87. Ke4 Ba7 88. Ra2 Bb6 89. Kf4 Bc7+,** Figure 5e, **½-½.**

6. TIVIAKOV-KORSUNSKY (1989)

Now finally comes a real revolution. Human theory has thought that Figure 6a's position 45w from Tiviakov-Korsunsky (Redhotpawn.com, 2014) is a fortress: the first author had also claimed this many times including (Müller, 2007). But White can win, as first pointed out by Jonathan Hawkins (2012) in his excellent book on page 105. Either White invades with his king to c6, this winning aim being known to human theory, or amazingly, White exchanges pawns with a3-a4 at the right time.

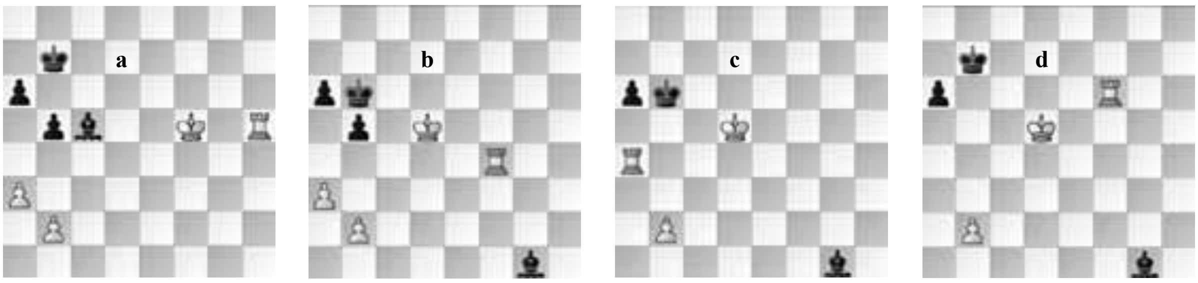


Figure 6. Tiviakov-Korsunsky: (a) 45w, and after sideline (b) 52. … Kb6, (c) 54. Rxa4 and (d) 56. … Kb7.

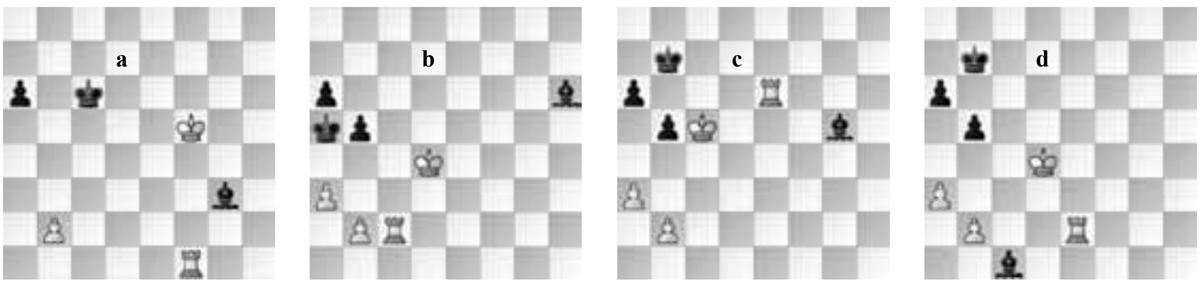


Figure 7. Tiviakov-Korsunsky after sideline (a) 62. Kf5, (b) 61. … Bh6, (c) 66. Kc5 and (d) 65. Re2.

45. Ke4 Bf2 46. Rf5 Bg1 47. Rf1 Bc5 48. Kd5 Be3 49. Rf7+ Kb6 50. Rf3 Bg1 51. Rf1 (51. Rf6+ Kb7 52. Rf4 Kb6, Figure 6b, is more direct. Now, remarkably, White should exchange pawns with 53. a4!! bxa4 54. Rxa4, Figure 6c, *dtc/m/z* = -50/-73/-41m. White's rook now wins a long domination duel as in, e.g., this initially DTC/M-minimaxing line from YK/AR: 54... Bf2′ 55. Rf4′ Bg1″ 56. Rf6+″ Kb7″ Figure 6d 57. Rf1″ Be3″ 58. Rf3″ Bg1″ 59. Kd6″ Bh2+″ 60. Ke6″ Kc6″ 61. Rf1″ Bg3″ 62. Kf5″ Figure 7a. This is really extraordinary! White's king has moved to f5 to win the domination fight. Chess really is a rich game! 62. … Bd6″ 63. Rc1+′ Kb6″ 64. Ke4″ Bc5″ 65. Kd3″ Kb5″ 66. Ra1″ Kb6″ 67. Kc4″ Be3″ 68. Re1′ Bf2″ 69. Rf1″ Be3″ 70. Rf3″ Bg1″ 71. Kb4″ Bd4″ 72. Rb3″ Be5″ 73. Ka4+″ Ka7″ 74. Ka5″ Bf6″ 75. Kb4″ Kb6″ 76. Ka4+″ Ka7′ 77.

Rb4″ SC^-/SC^+ Bd8 (*SM^-C^-/SM^+C^+: 77. ... Be5″ 78. Kb3″ Bd6″ 79. Rg4″ Be5″ 80. Re4″ Bg3″ 81. Kb4″ Kb6″ 82. Rg4″ Bb8″ 83. Kc4″ Kc6′ 84. Rg6+″ Kb7″ 85. Kd5″ Bf4″ 86. Rg4″ Bb8″ 87. Kc5″ Ba7+″ 88. Kd6″ Bb8+″ 89. Kd7″ a5″ 90. Rc4′ Kb6″ 91. Ke6″ Kb5″ 92. Kd5″ a4″ 93. Rc5+″ Kb6″ 94. Kc4″ Bf4″ 95. Rb5+″ Ka6″ 96. Kc5″ Be3+″ 97. Kc6″ Bc1″ 98. Rb8″ Ka5″ 99. Kc5″ Be3+″ 100. Kc4″ Bd2′ 101. Ra8+″ Kb6° 102. Rxa4″* +-) 78. Kb3″ Ba5″ 79. Rg4″ Kb7″ 80. Kc4″ Bb6″ 81. Kd5″ Bf2″ 82. Kd6″ Be1″ 83. Rg8″ Bb4+′ 84. Kd5″ Be1″ 85. Rf8′ Kb6″ 86. Rf6+′ Kb5″ 87. Rf4″ Bd2″ 88. Rf8″ Kb6″ 89. Rb8+″ Kc7″ 90. Re8″ Bc1″ 91. Re2″ Kb6″ 92. Kc4″ Bf4′ 93. Re6+″ Kb7′ 94. Kc5″ Bg5″ 95. Rb6+″ Ka7″ 96. Kc6″ Be3″ 97. Rb7+′ Ka8° 98. b4′ Bd4′ 99. Rd7″ Bf2″ 100. Rd2″ Be1″ 101. Rd1″ Bf2″ 102. Ra1″ Ka7″ 103. b5″ a5′ 104. Rxa5+″ +-)

51. ... Be3 52. Ke4 Bg5 53. Rf5 Bc1 54. Rf2 Bg5 55. Kd4 Bc1 56. Re2 Ka5 (56. ... Bg5 57. Re6+ Kb7 58. Kc5 Bd8 59. b4 Bh4 60. Rb6+ Ka7 61. Kc6+-)

57. Kc3 Kb6 58. Kd4 Ka5 59. Rc2 Bh6 60. Rg2 Bc1 61. Rc2 Bh6 Figure 7b **62. Rc7?!** allows Black to get back in his house. (62. Rg2 wins more quickly, e.g., 62. … Bc1 (*62. ... Bf8 63. Kc3 Kb6 64. Rg6+ Kb7 65. b4*+-) (*62. ... Ka4 63. Rg6 Bc1 64. Kc3*+-) 63. Re2 Kb6 (*63. ... Bh6 64. Kc3 Bg7+ 65. Kb3 Bf6 66. Re6 Bd4 67. Ka2 b4 68. axb4+ Kb5 69. Kb3*+-) 64. Kd5 Bg5 (*64. ... a5 65. Kd4 a4 66. Kd5 b4 67. Rc2 Be3 68. axb4 Kb5 69. Rc8*+-) 65. Re6+ Kb7 66. Kc5, Figure 7c, and White's king invades to c6. 66. … Bd8 67. b3 Bg5 68. Rb6+ Ka7 69. Kc6+-)

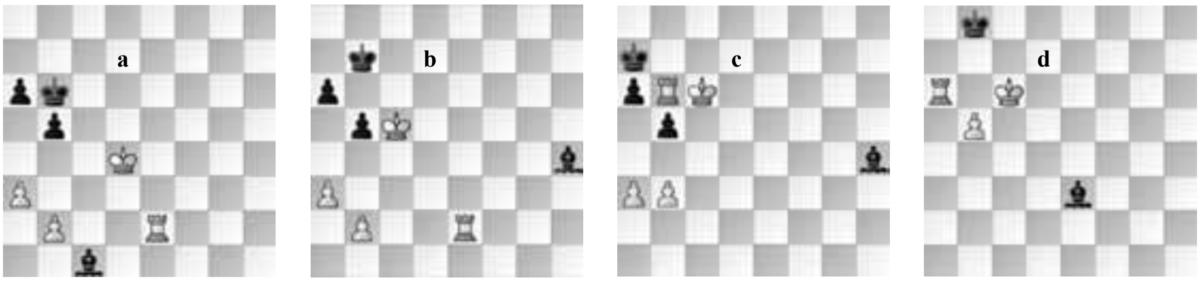


Figure 8. Tiviakov-Korsunsky after (a) 65. … Kb6?!, (b) 66. … Bh4, (c) 71. Kc6, and (d) 79. Rxa6.

62. ... Kb6 63. Re7 Bc1 (63. ... Bg5 64. Re6+ Kb7 65. Kc5 Bd8 66. b3 Bh4 67. Rb6+ Ka7 68. Kc6+-) **64. Re6+ Kb7 65. Re2**, Figure 7d. Even 65. Kc5 is playable. 65. … Bxb2 66. Re7+ Kb8 67. Re3 (*67. Kb6? Bd4+ 68. Kxa6 Bc5*=) 67. ... Kc7 68. Rf3 Kb7 69. Rh3 Kc7 70. Rh7+ Kb8 71. Kb6+-) **65. ... Kb6**?!, Figure 8a, and now the bishop is dominated. (65. ... Bg5!? 66. Kc5 Bh4, Figure 8b, was more tenacious, e.g., 67. a4 bxa4 68. Kb4 Bg3 69. Kxa4, *dtc/m/z* = -53/-76/-44m, and as in, e.g., this DTC/M-minimaxing line from YK 69... Bc7″ 70. Re6′ Bd8′ 71. Kb4′ Bb6″ 72. Kc4″ Bg1″ 73. Rf6′ Be3′ 74. Kd5″ Bg1″ +-, Figure 6d once again)

66. Kd5 Bg5 (66. ... a5 67. Kd4 a4 68. Kd5 b4 69. Rc2 Be3 (*69. ... bxa3 70. Rxc1 axb2 71. Rb1 a3 72. Kc4*+-) 70. axb4 Kb5 71. Rc8+-) (66. ... Ka5 67. Rc2 Be3 (*67. ... Bf4 68. Rc6*+-) 68. Rc6 b4 69. axb4+ Kb5 70. Rc8 Bf4 71. Rc5+ Kb6 72. Kc4+-)

67. Re6+ Kb7 68. Kc5 Bd8 (68. ... Bh4 69. Rb6+ Ka7 70. Kc6+-) **69. b3 Bh4 70. Rb6+″″ Ka7 71. Kc6**, Figure 8c. White's king has reached the key square c6 and it is over. **71. … Bf2 72. Rb7+″″ Ka8° 73. Rf7 Bg1 74. Rf4 Ka7 75. a4 bxa4 76. Rxa4** KRPKBP, *dtc/m* = -7/-12m **Bf2 77. b4″ Be3 78. b5″ Kb8″ 79. Rxa6″**, Figure 8d, **1-0.**

7. SUMMARY

The EGTs show that the defending side has less scope to draw than previously thought. It is for example not enough to hold the main fortress from Gelfand-Ivanchuk by just waiting with the bishop on the long diagonal and the structure from Tiviakov-Korsunsky can surprisingly be won in a long domination duel by the rook, which even current engines do not find and which can only be revealed by the EGTs. Chess really is a very deep game and we have much to learn, especially when more pawns appear on the board. Further study will be assisted by the accompanying pgn file, its light annotation and the FREEZER KRPKBP EGTs (Müller and Haworth, 2014). Recommended sources include Chéron (1969), Timman (1996) and Müller (2012).

Thanks go to Eugene Nalimov for his sub-7-man DTM EGTs, to Eiko Bleicher for his EGT-query service and FREEZER software, to Árpád Rusz for his FREEZER EGT results, to Yakov Konoval for his 6-man DTC EGTs and results, to P.P.Romero for FINALGEN, to the Lomonosov team for their 7-man DTM EGTs and to Harold van der Heijden for some DTZ depths.

References

Bleicher, E. (2014a). Nalimov EGT look-up service http://www.k4it.de/index.php?topic=egtb&lang=en.

Bleicher, E. (2014b). FREEZER. http://www.freezerchess.com/.

Chéron, A. (1969). *Lehr und Handbuch Der Endspiele*. Vol. 1, pp. 315-329 (esp. p. 323) and Vol. 3, pp. 285-297. Siegfried Engelhardt Verlag. ASIN B0021WO8L2.

Chessgames.com (2014a). Sachdev-Schut. http://www.chessgames.com/perl/chessgame?gid=1654450.

Chessgames.com (2014b). Timman-Velimirović. http://www.chessgames.com/perl/chessgame?gid=1142548.

Chessgames.com (2014c). Gelfand-Ivanchuk. http://www.chessgames.com/perl/chessgame?gid=1610267.

ChessOK (2014a). Online access to the sub-7-man Nalimov EGTs. http://chessok.com/?page_id=361.

ChessOK (2014b). Online access to the MVL 7-man DTM EGTs. http://chessok.com/?page_id=27966.

Donner, J.H. (2007). *The King: Chess Pieces*, pp. 319-326. Interchess BV. ISBN 9-0569-1171-6.

Dvoretsky, M. (2003). *Dvoretsky's Endgame Manual.* 3rd edition (2011). Russell. ISBN 978-19-3649-013-4.

Hawkins, J. (2012). *Amateur to IM: Proven Ideas and Training Methods*. Mongoose Press. ISBN 1-9362-7740-9.

Herik, H.J. van den, Herschberg, I.S. and Nakad, N. (1987). A Six-Men-Endgame Database: KRP(a2)KbBP(a3). *ICCA Journal*, Vol. 10, No. 4, pp. 163-180.

Herik, H.J. van den, Herschberg, I.S. and Nakad, N. (1988a). Karpov Amends Timman's Analysis. *ICCA Journal*, Vol. 11, No. 1, pp. 32-33.

Herik, H.J. van den, Herschberg, I.S. and Nakad, N. (1988b). A Reply to R. Sattler's Remarks on the KRP(a2)-KbBP(a3) Database. *ICCA Journal, Vol. 11*, Nos. 2/3, pp. 88-91.

Konoval, Y. (2014). Private communications of 6-man DTC information.

Müller, K. (2007). Endgame Corner #70. http://www.chesscafe.com/text/mueller70.pdf.

Müller, K. (2012). Chess Endings 11: Rook vs Bishop. ChessBase Fritztrainer DVD, Hamburg.

Müller, K. (2013). Endgame Corner #143. http://preview.tinyurl.com/pjgxewz.

Müller, K. and Haworth, G.McC. (2014). Rook versus Bishop. This paper plus pgn file, FREEZER EGTS and further positions and lines. http://centaur.reading.ac.uk/36189/.

Müller, K. and Lamprecht, F. (2001). *Fundamental Chess Endings*, §7.2.C1 pp. 274-8 on KRP(x2)KBP(x3), especially example 7.42 on Timman-Velimirović. Gambit. ISBN 1-9019-8353-6.

MVL (2012). Blog on the MVL EGT project. https://plus.google.com/100454521496393505718/posts.

Nunn, J. (1981). *Tactical Chess Endings*, esp. pp. 190-7, #134-134e. George Allen and Unwin.

Nunn, J. (2002). *Secrets of Pawnless Endings*, esp. pp. 31-48, Ch.2.2 and #38. Gambit ISBN 1-9019-8365-X.

Nunn, J. (2005). Recent advances in endgame theory. *New in Chess*, Issue 2005/6, pp. 71-9.

Redhotpawn.com (2014). Tiviakov-Korsunsky. http://preview.tinyurl.com/onhykx7.

Romero, P. P. (2012). FINALGEN: download, tutorial and examples. http://www.mtu-media.com/finalgen.

Roycroft, A.J. (1983). Chess Games, July 30, 1983 *C*, A Prophecy Fulfilled. EG, Vol. V, No. 74, pp. 217-220. See also http://www.gadycosteff.com/eg/eg74.pdf.

Roycroft, A.J. (1988). Expert against the Oracle. Machine Intelligence 11 (eds. J.E. Hayes, D. Michie and J. Richards) pp. 347-373. Oxford University Press, Oxford.

Rusz, Á. (2014). Private communications of FREEZER KRPKBP EGTs and results.

Sattler, R. (1988). Further to the KRP(a2)KbBP(a3) database. *ICCA Journal*, Vol. 11, Nos. 2/3, pp. 82-87.

Timman, J. (1981). *Schaakbulletin 166*. http://www.svtheothorne.nl/bibliotheek.php.

Timman, J. (1996). *Studies and Games*, esp. pp. 25-34 and 127-43. Everyman. ISBN 1-8574-4126-0.

Timman, J. (2011). *The Art of the Endgame*, esp. pp. 22-3. New in Chess. ISBN 978-90-5691-369-4.

Van der Heijden, H.M.J.F. (2010). http://www.hhdbiv.nl/. HHDBIV, ENDGAME STUDY DATABASE IV, #4.

APPENDIX

The maxDTC KRP(a2)KBP(a3) win (Figure 2e), 1K6/8/1k6/6R1/8/p3b3/P7/8 w, *dtc/m* = 55/82m …
SC$^-$/SC$^+$: 1. Rd5⁗ Bc1″ 2. Rd8⁗ Bb2″ 3. Rc8⁗ Be5+″ 4. Ka8⁗ Bc7″ 5. Re8″ Bf4″ 6. Re4″ Bc1″ 7. Rc4″ Bb2″ 8. Kb8″ Kb5″ 9. Rc2″ Kb4″ 10. Rc6″ Bd4″ 11. Kc7″ Bc5″ 12. Ra6⁗ **{and here, SM$^+$ diverges}** Kc3′ 13. Kc6⁗ Kc4″ 14. Ra4+′ Bb4″ 15. Kb6″ Kc3″ 16. Kb5″ Bf8″ 17. Ra8′ Bd6′ 18. Re8′ Kb2″ 19. Re2+⁗ Kc3″ 20. Ka4″ Bc5′ 21. Re5″ Bd6′ 22. Rb5″ Be7″ 23. Rb1′ Bd6′ 24. Rb3+″ Kc4″ 25. Rb6″ Be7′ 26. Rc6+″ Bc5″ 27. Rc7′ Kd4″ 28. Kb3″ Kd5″ 29. Rf7′ Bd6′ 30. Rf5+′ Kd4′ 31. Rf1′ Kd5′ 32. Rd1+″ Ke6″ 33. Rd3″ Ke5′ 34. Kc4″ Bf8″ 35. Rf3′ Bd6″ 36. Rf2″ Ke6″ 37. Re2+″ Kf6″ 38. Kd5′ Bb4″ 39. Re3′ Kf5″ 40. Rf3+″ Kg4″ 41. Rb3′ Bf8″ 42. Rb7′ Bh6″ 43. Rc7″ Kh5″ 44. Ke6″ Kg6″ 45. Rc3″ Bf8″ 46. Rg3+″ Kh5′ 47. Kf6″ Kh4′ 48. Rc3″ Kh5″ 49. Rc4′ Bh6″ 50. Kf5″ Bf8″ 51. Rc8″ Bg7″ 52. Rc1″ Kh4″ 53. Rh1+″ Kg3° 54. Rg1+″ Kf3′ 55. Rxg7⁗ **{KRPKP: *dtc/m/z* = -2/-9/-2m} {YK DTC EGT (Konoval, 2014), Freezer DTC EGT (Rusz, 2014)}** 1-0.

The maxDTC KRPKBP loss, 8/6p1/8/8/1b6/2k5/6P1/3K2R1 b, *dtc/m* = -99/-121m (Konoval, 2014) …
SC$^-$/SC$^+$: 1. ... Kd3″ {the maxDTC KRPKBP wtm win} 2. Rh1⁗ Bd6″ 3. Re1⁗ Kd4″ 4. Ke2⁗ Ke4″ 5. Kf2+⁗ Kf4″ 6. Rc1⁗ Be7″ 7. Rc4+⁗ Kg5′ 8. Rc6⁗ Kf4″ 9. Kg1‴ Bf6″ 10. Kh2⁗ Kg4″ 11. Rc5⁗ {zugzwang} Be7″ 12. Rc4+″ Kf5″ 13. Rc6⁗ Bh4″ 14. Kh3⁗ Be1″ 15. Rc1⁗ Bd2″ 16. Rc2″ Be3″ 17. Kh4″ Bg5+″ 18. Kg3″ Bh6′ 19. Rc6‴ Kg5″ 20. Rd6′ g6″ 21. Rc6′ Bg7′ 22. Kf3′ Kf5″ 23. Ke3′ Bh8′ 24. Kd3″ Bg7′ 25. Kc4″ Bh8′ 26. Kb5″ Be5′ 27. Kc5″ Bf4′ 28. Kd5″ Bh6″ 29. Rc3′ Kf6″ 30. Kd6⁗ Bf4+″ 31. Kd7″ Be5″ 32. Re3⁗ Bd4′ 33. Re7″ Bc5″ 34. Rh7⁗ Be3″ 35. Ke8″ Bf4″ 36. Rd7′ Kf5″ 37. Rd5+′ Kg4″ 38. Rd3⁗ g5′ 39. Rf3′ Be5′ 40. Kf7′ Bh8′ 41. Ke6″ Bd4′ 42. Kd5″ Bg7′ 43. Ke4″ Bh8′ 44. Ke3′ Bg7′ 45. Ke2″ Bh8′ 46. Rd3″ Kf4″ 47. Kf2″ g4″ 48. Rd7′ Bc3″ 49. Rf7+″ Kg5″ 50. Re7′ Kf4″ 51. Rd7″ Be5″ 52. Rf7+″ Kg5″ 53. Rb7″ Bd6″ 54. Rb5+″ Kh4″ 55. Rb3″ Bc7′ 56. Ke2″ Bd8′ 57. Kd2″ Be7′ 58. Rd3′ Bb4+″ 59. Kc2″ Bc5′ 60. Kc3′ Bf2″ 61. Kc4″ Kg5″ 62. Kd5″ Kf5″ 63. Rc3′ Bh4″ 64. Rc4″ Bg5′ 65. Re4″ Bh4″ 66. Kd4″ Bg3″ 67. Ke3′ Bf2+″ 68. Kd3⁗ Bh4′ 69. Rc4′ Bf2″ 70. Ke2″ Bh4″ 71. Rd4″ Bf6″ 72. Rd7″ Ke4″ 73. Rc7′ Bh4′ 74. Rf7″ Bg3″ 75. Kd2″ Be5″ 76. Kc2″ Kd4′ 77. Kb3″ Kd5″ 78. Rf1′ Bc7′ 79. Kb4′ Bb8′ 80. Re1′ Be5′ 81. Kb5″ Bd6′ 82. Rd1+′ Ke5″ 83. Kc6″ Bb8″ 84. Rd8″ Ba7″ 85. Rf8″ Be3″ 86. Rf1′ g3′ 87. Re1′ Kf4″ 88. Kd5″ Bb6″ 89. Re4+′ Kg5′ 90. Re5+″ Kg4′ 91. Ke6″ Bc7′ 92. Rf5′ Bd8′ 93. Rb5′ Bg5′ 94. Rb4+″ Kh5″ 95. Re4″ Bh6′ 96. Kf5″ Bd2″ 97. Re7′ Bg5′ 98. Rg7′ Kh4″ 99. Rh7+″ Bh6° 100. Rxh6#″ **{YK DTC EGT (Konoval, 2014), Freezer DTC EGT (Rusz, 2014)}** 1-0.

The maxDTM KRPKBP wtm win, 8/3R2P1/7k/8/8/8/5p2/K5b1 w, *dtc/m/z* = 1/166/1m …
SM$^-$/SM$^+$: 1. g8=N+⁗ **{KRNKBP: *dtc/m/z* = -5/-165/-5m}** Kg6″ 2. Rd1⁗ Kf5′ 3. Kb2′ Ke4′ 4. Kc2⁗ Ke3″ 5. Nf6⁗ Ke2″ 6. Ne4⁗ f1=N″ **{KRNKBN: *dtm* = 159m}** 7. Nc3+⁗ Kf2″ 8. Rc1′ Ng3′ 9. Kd3⁗ Kg2″ 10. Nd5⁗ Ba7″ 11. Rc7⁗ Bb8′ 12. Rf7″ Kh3″ 13. Rf8⁗ Be5″ 14. Re8⁗ Bd6″ 15. Re6″ Bc5″ 16. Rc6″ Ba7″ 17. Rc4″ Nh5″ 18. Rc7″ Bf2″ 19. Rc8″ Kg4″ 20. Rg8+⁗ Kf3″ 21. Rf8+⁗ Kg2″ 22. Rf5″ Ng3″ 23. Rf6′ Nh5″ 24. Rf8″ Bg1″ 25. Rg8+′ Ng3″ 26. Nf4+″ Kf3″ 27. Ne6⁗ Ba7″ 28. Ra8⁗ Bb6″ 29. Rb8″ Ba7″ 30. Rb7‴ Bg1″ 31. Rf7+″ Kg4″ 32. Rg7+″ Kh4″ 33. Rg8′ Bf2′ 34. Nf8″ Bg1′ 35. Nd7′ Kh3″ 36. Rg6″ Bf2″ 37. Rf6″ Bg1′ 38. Rf4″ Nh5″ 39. Ra4″ Bf2″ 40. Ne5″ Bg3″ 41. Ng6⁗ Bf2″ 42. Rc4″ Kg2″ 43. Ne5″ Bg3″ 44. Nf7⁗ Bb8″ 45. Rc8″ Ba7″ 46. Rh8″ Nf4+″ 47. Ke4⁗ Ne2″ 48. Ng5″ Bb6″ 49. Rf8″ Ng3+″ 50. Kd3″ Nh5′ 51. Rf5″ Ng3′ 52. Rb5″ Bf2″ 53. Rb7′ Nf5′ 54. Rb2″ Ng3″ 55. Rc2″ Nh1″ 56. Ne6″ Ng3″ 57. Nd4″ Nh1″ 58. Ke4″ Kh3″ 59. Nc6⁗ Kg3″ 60. Ne5″ Bg1″ 61. Rc7″ Nf2+″ 62. Kd5″ Nh1″ 63. Rf7″ Bb6″ 64. Rh7″ Kg2″ 65. Nc4⁗ Bg1″ 66. Rf7″ Ng3″ 67. Ke5⁗ Nf1″ 68. Kf4″ Bh2+″ 69. Kg4″ Bg3″ 70. Rb7′ Nh2+″ 71. Kf5″ Nf1″ 72. Ke4″ Bh4′ 73. Rg7+″ Kf2″ 74. Kd3″ Kf3″ 75. Ne5+″ Kf4″ 76. Ng6+⁗ Kg4″ 77. Nf8+‴ Kf3″ 78. Ne6′ Ng3″ 79. Rf7+⁗ Kg2″ 80. Ke3″ Nf1+″ 81. Ke4″ Ng3+″ 82. Kf4″ Nf1″ 83. Ke5″ Nd2′ 84. Rd7″ Bg3+″ 85. Kd4″ Nf1″ 86. Rg7″ Kf2″ 87. Kd3″ Be5′ 88. Rg5″ Bd6″ 89. Rd5″ Bh2″ 90. Rb5″ Kf3″ 91. Rb4″ Be5″ 92. Re4″ Bb2″ 93. Rf4+″ Kg2″ 94. Ra4″ Bh8″ 95. Ra6′ Be5″ 96. Ke4″ Bg3″ 97. Ra7″ Nd2+′ 98. Kd3⁗ Nf3″ 99. Rg7″ Kh3″ 100. Ke3″ Nh2′ 101. Ke4″ Nf1′ 102. Rg8″ Nd2+′ 103. Kd3′ Nf1′ 104. Ke2″ Nh2″ 105. Nd4″ Bf4′ 106. Nf5″ Be5′ 107. Ke3′ Bc7″ 108. Rc8″ Be5″ 109. Rc4″ Kg2″ 110. Ke2″ Bf6″ 111. Rc6″ Be5″ 112. Rg6+″ Kh3″ 113. Kf2″ Bc7″ 114. Rg7″ Bf4″ 115. Rg8″ Be5″ 116. Rg5″ Bf4″ 117. Rg7″ Be5″ 118. Rb7″ Kg4″ 119. Ne3+⁗ Kg5″ 120. Rb5⁗ Kf4″ 121. Rb4+⁗ Kg5″ 122. Kg2‴ Kg6″ 123. Rc4′ Kf7′ 124. Ra4″ Kg6″ 125. Ra6+″ Kf7′ 126. Ra5″ Kf6″ 127. Rd5″ Bb8′ 128. Rd8″ Be5″ 129. Rf8+″ Ke7″ 130. Rf2″ Kd7″ 131. Rd2+″ Ke6′ 132. Re2″ Kf6″ 133. Rf2+′ Kg6″ 134. Kh3″ Bd4″ 135. Rg2+⁗ Kh5′ 136. Nd5⁗ Be5″ 137. Rf2⁗ Kg5″ 138. Rf8″ Bd6″ 139. Rf7″ Be5″ 140. Ne7′ Bb8″ 141. Rf8′ Bc7″ 142. Rf5+″ Kh6° 143. Rf2″ Kg5′ 144. Rf8″ Kh6′ 145. Nd5″ Be5′ 146. Re8″ Bd6″ 147. Re6+″ Kg5″ 148. Rxd6″ **{KRNKN: *dtc/m/z* = -12/-18/-12m}** Nf3″ 149. Kg3′ Nd2″ 150. Nf6″ Nc4′ 151. Rd4″ Ne3′ 152. Kf3′ Nf5′ 153. Rf4″ Ne7′ 154. Ke4″ Ng6′ 155. Rf1′ Ne7′ 156. Nd5″ Nc6′ 157. Rg1+′ Kh5″ 158. Rg2′ Kh4″ 159. Rg6′ Nd8′ 160. Nf4′ Nc6″ 161. Rxc6″ **{KRNK: *dtc/m/z* = -4/-5/-4m}** Kg5′ 162. Kf3′ Kf5″ 163. Ng6″ Kg5° 164. Nh4′ Kh5′ 165. Kf4″ Kxh4° **{KRK: *dtx* = 1m}** 166. Rh6#″ **{Nalimov DTM EGTs}** 1-0.

Multi-Level Inference in Chinese Chess Endgame Knowledge Bases

Bo-Nian Chen[1], *Hung-Jui Chang*[2], *Shun-Chin Hsu*[3], *Jr-Chang Chen*[4], *and Tsan-sheng Hsu*[5]

ABSTRACT

In Chinese chess, retrograde analysis can be used to solve complex elementary (i.e., fundamental) endgames and to provide perfect play. However, there are still many practical endgames pending to be analysed due to problems related to the complex playing rules. Of course, there is heuristic endgame knowledge for the evaluation functions. This knowledge is often applied to the complex endgames or the real endgames to improve the playing strength. One crucial problem is to choose relatively advantageous endgames by selecting appropriate piece exchanges. For this problem, we designed a Chinese chess endgame knowledge-based system with a large set of endgame heuristics, called an *endgame knowledge base*. We use this knowledge base in our program, CONTEMPLATION. To maintain the quality of the constructed knowledge base, it is important to detect and resolve conflicts between the heuristics. A conflict-resolution method enables Chinese chess experts to correct erroneous entries by using knowledge about two endgames that differ by precisely one piece. The problem involves detecting potential errors so that a human expert can easily revise and improve the reliability of the knowledge base. In this article, we introduce two major enhancements to the above method. First, we propose a general graph model to handle the heuristics when the endgames involved are differing in more than one piece. Second, we add a confidence-factor parameter to encode the probability that a heuristic may be true. Such heuristics are often used in real games when pieces are exchanged. The resulting graph model is effective in maintaining the consistency of predefined meta-knowledge, and thus improves the overall quality significantly. The results of the experiments on self-play tests demonstrate that the derived knowledge base improves the playing strength of CONTEMPLATION.

1. INTRODUCTION

Chinese chess is a challenging two-player zero-sum perfect-information game. It is played by humans as well as by computers. According to Van den Herik, Uiterwijk, and Van Rijswijck (2002) the state-space complexity is 10^{48} and the game-tree complexity is 10^{150}. The corresponding complexities of Checkers, a game solved by Schaeffer *et al.* (2007) are 10^{21} and 10^{31} respectively. We believe that Chinese chess will not be solved in the near future.

Two main aims are improving the playing strength and developing new techniques for adequately handling large amounts of data. There are several knowledge-based systems that support achieving the two aims in a variety of computer games. For complete information games, Ciancarini and Gaspari (1989) designed a knowledge-based system that uses evaluation functions as rules to generate plans in middle game positions in Chess. Rubin *et al.* (2008) proposed a principle that uses a knowledge base with complex basic knowledge items in combination with an appropriate knowledge-acquisition procedure to build a dependence space of knowledge items for Chess. For incomplete information games, the rational approach to combine substantial and procedural rationality was proposed for Kriegspiel (Ciancarini and Dalla Libera, 1997). Knowledge-based systems are also often applied in Chinese chess. Chen and Hsu (2001) proposed an automatic method for constructing an opening game knowledge base by analyzing a large number of games. To enhance the playing strength in the endgame, Wu, Liu, and Hsu

[1]Institute of Information Science, Academia Sinica, Taipei, Taiwan. email:brain@iis.sinica.edu.tw
[2]Institute of Information Science, Academia Sinica, Taipei, Taiwan. email:changhungjui@gmail.com
[3]Department of Information Management, Chang Jung Christian University, Tainan, Taiwan. email:schsu@mail.cjcu.edu.tw
[4]Department of Applied Mathematics, Chung Yuan Christian University, Taoyuan, Taiwan. email:jcchen@cycu.edu.tw
[5]Corresponding author. Institute of Information Science, Academia Sinica, Taipei, Taiwan. email:tshsu@iis.sinica.edu.tw

(2006) designed a retrograde algorithm that uses an external memory to solve all complex elementary endgames (we call them “fundamental” endgames).

In Chinese chess, there are still many endgames that are currently not possible to be solved by using retrograde analysis. Nevertheless, these kinds of endgames can be handled by a program, when we use *endgame heuristics* to build an effective endgame evaluation function. In the transition from the middle game to the endgame, a major problem is to find the way to select an appropriate endgame by suitable piece exchanges. To achieve the best endgame, we proposed an automatic system, see (Chen *et al.*, 2008), that manually aggregated endgame knowledge mainly consisting of heuristics, and implemented an inference technique to obtain automatically a large number of knowledge rules. Since automatic generated heuristics may contain errors, the errors may cause a Chinese chess program to be unstable. We improved the system with a conflict-detection algorithm and a repair mechanism, which helped us compile a consistent endgame knowledge base of about 70 thousand knowledge rules (Chen *et al.*, 2009). Then, we introduced the lattice model to refine our knowledge base and expand it to approximately 120 thousand knowledge rules (Chen *et al.*, 2012). In this article, we propose the *distance-k graph model* to improve the quality of the previous knowledge base and create a larger knowledge base with a higher quality, that consists of about 140 thousand knowledge rules. The new inference model contains critical information about the exchange of pieces, which is important in the transition from the middle game to the endgame.

The techniques used in the knowledge-based system we currently propose will have four similarities with the techniques proposed by Rubin *et al.* (2008), viz. (1) they can both acquire knowledge and perform inferences on knowledge bases, (2) they both build knowledge bases from fundamental (i.e., complex elementary) knowledge, (3) the knowledge bases are created with the help of human experts, and (4) both types of knowledge bases are heuristically oriented. However, there are also three differences between the two systems: (1) the former knowledge base is applied to a minimax algorithm while the latter is applied to a goal-oriented algorithm, (2) the former knowledge-based system automatically generates approximate knowledge and then correct the conflicts by a conflict resolution algorithm, and (3) the relations among the data of the former knowledge base is flexible while those of the latter is implied by the input process. The flexible relations make our knowledge-based system suitable for a self-validation process and thus reduce the human efforts during construction.

The remainder of this paper is organized as follows. In Section 2, we define the elements in our knowledge-based system and present the corresponding graph models. In Section 3, we describe the proposed generalized conflict-resolution algorithm; and in Section 4, we discuss the results of our experiments. Section 5 contains four concluding remarks.

2. THE GRAPH MODEL FOR ENDGAME KNOWLEDGE BASES

The heuristic used to describe an endgame contains the piece types in the given endgame together with a score (see 2.1). The next step is to transform the heuristics into a graph that must maintain the consistency among the heuristics. For negative cases, we designed a method to detect and resolve conflicts based on the graph model. In this section, we describe the method used to implement the heuristics in an endgame knowledge base (see 2.2) and the mechanism that ensures the consistency of our knowledge base (see 2.3).

2.1 Material Combinations

A *material combination* is defined according to the pieces that are on a board. We denote the seven pieces used in Chinese chess as follows: king (K), guard (G), minister (M), rook (R), knight (N), cannon (C), and pawn (P). A material combination can be divided into the three parts: *strong pieces*, pawns, and *defending pieces*. Rooks, knights, and cannons are strong pieces; while guards and ministers are defending pieces. We denote a material combination as a string of “red pieces followed by black pieces”. For example, KCPKGGMM denotes the material combination that the red side has a cannon and a pawn and the black side has two guards and two ministers. A Chinese chess expert can evaluate an endgame position only based on his knowledge of the material combination of that endgame. For a given material combination, we assign an *advantage score* to indicate the advantage of the set of positions with the same material combinations from the perspective of the red player. The score comprises 12 levels, from 0 to 11. The following scores give the red side an advantage: 0 (certain win), 1 (probable win), 2 (advantage), 3 (slight advantage with a chance of winning), and 4 (slight advantage, but unlikely to win). The advantage score is 5 (tie) if both sides have an equal chance of winning, and 6 (draw) if either side has little chance of winning. The scores 7, 8, 9, 10, and 11 are the opposites of 4, 3, 2, 1, and 0 respectively. The advantage score in Chen *et al.* (2012) only contains 10 levels. In this article, we have added two levels, 4 and 7, to help the players distinguish between positions that could result in a win and those that can only lead to a draw.

Because the value of a material combination is relative (cf. Chen *et al.*, 2012), it is difficult to determine the values of all endgame material combinations consistently. To address the issue, we propose a graph model that captures the rationality of the relations among the material combinations in Chinese chess.

2.2 The Graph Model

The heuristics used contain information of material combinations. Each material combination is represented as a node in a graph. The heuristics are verified. This happens by meta-knowledge rules about a relation between two heuristics. According to the meta-knowledge rules that we define, we can detect and try to resolve conflicting heuristics for the cases that the difference between two material combinations is more than one piece, especially for piece exchanges.

Definition 1 *Let x be a node that represents a material combination, and let $x.m$ be the set of pieces in x. The set contains red and black pieces, denoted as $x.m.r$ and $x.m.b$ respectively; and its advantage score is denoted as $x.v$. For two adjacent nodes x and y, a directed edge $x \to y$ indicates that the red side has at least the same advantage in x as it has in y.*

Our model contains a directed graph Γ in which each node represents a material combination and each edge represents the relation between two material combinations x and y. In the directed graph, $x \to y$ implies that x and y are directly related and the advantage score of y cannot be better than that of x. A *conflict* between x and y means that the piece types and the advantage scores of x and y violate some of the meta-knowledge rules about the Chinese chess endgame. If the two nodes follow all the meta-knowledge rules, they are said to be *consistent*.

Definition 2 *Let $|x.m|$ denote the number of pieces in $x.m$; and let $x.m - y.m$ denote the pieces that are in $x.m$ but not in $y.m$. Then, we define the "distance" between x and y as $d(x, y) = |(x.m - y.m) \cup (y.m - x.m)|$.*

Definition 3 *The set of distance-k neighbor nodes of node x is defined as follows:*

$$N_k(x) = \{y \mid \textit{where } 0 < d(x, y) \leq k\}.$$

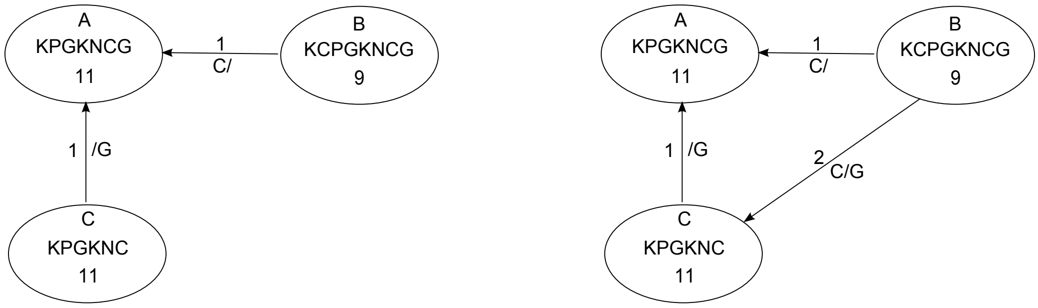
Definition 4 *The distance-k graph model $\Pi_k = (\Gamma, D_k, \Phi)$ comprises a directed graph Γ with two functions, $D_k(x)$ and $\Phi(x, y)$. The neighbor generator D_k computes $N_k(x)$, where $x \in \Gamma$. The conflict detector Φ determines if node x conflicts with its neighboring node y. The output of Φ is either true (conflict) or false (consistent).*

In the formula, the conflict detector Φ may contain Chinese chess knowledge that defines conflicts between adjacent nodes. In Chen *et al.* (2012), the piece-additive rule is introduced saying that you cannot lose the level of advantage by capturing some opponent's pieces. In another paper (Chen *et al.*, 2014), we will introduce more knowledge in larger distance-d graphs. We incorporate the Φ function into a conflict resolution algorithm, which is described in Section 3. The method enables us to compile a consistent knowledge base.

Definition 5 *A consistent graph Γ^* is a graph in which there is no conflict between any pair of adjacent nodes.*

According to Definition 4, Π_1 is a special case called the *distance*-1 *graph model*, which is equivalent to the lattice model defined in Chen *et al.* (2012). The formula for the distance-1 graph model is $\Pi_1 = (\Gamma, D_1, \Phi)$, where the neighbor generator $D_1(x)$ collects the distance-1 neighbor nodes. Note that in Π_1, $x \to y$ in Γ occurs when (1) $x.m.r \supset y.m.r$ and $x.m.b = y.m.b$, or (2) $x.m.r = y.m.r$ and $x.m.b \subset y.m.b$.

In real games, pieces are exchanged during the transition from the middle game to the endgame. In Figure 1 (a), node B becomes node A after removing a red cannon, and node A becomes node C after removing a black guard; that is, node B becomes node C after exchanging a red cannon for a black guard. Because there is no edge between B and C in the distance-1 graph model, the original model cannot provide information about piece exchanges and thus may not identify potential conflicts.



(a) distance-1 graph model (b) distance-2 graph model

Figure 1: An example of the relationships between exchanged pieces. The numerical value on each node indicates its advantage score. The values on the edges represent the distances to the endpoints. The different pieces between two nodes are denoted by $r_1r_2...r_m/b_1b_2...b_n$. The following figures in this paper use this representation as well.

2.3 The Distance-2 Graph Model

The larger-distance graph model provides more ways to find subtle potential errors. In the case where $k = 2$, $\Pi_2 = (\Gamma, D_2, \Phi)$ is called the *distance*-2 *graph model*. There is an extra edge between node B and node C in Π_2, as shown by the example in Figure 1(b). Hence, this model can handle conflicts occurred by piece exchanges.

Note that the distance-2 graph model contains information about the exchange of one piece by each side. Assume there are three nodes x, y, and z. Figure 2 shows three types of inferences. In practice, we assume $|x.m| > |y.m| > |z.m|$. Type a contains two transitive inferences. Type b means that $x.m.r \supset y.m.r$, $x.m.b = y.m.b$, $y.m.r = z.m.r$ and $y.m.b \supset z.m.b$. Type c means that $x.m.r = y.m.r$, $x.m.b \supset y.m.b$, $y.m.r \supset z.m.r$ and $y.m.b = z.m.b$.

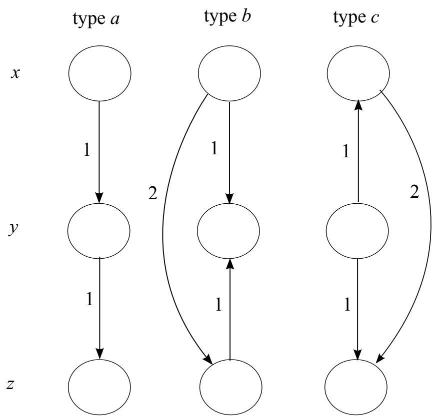


Figure 2: Three types of inferences in a distance-2 graph model.

We can create distance-2 inferences to find more potential errors. Because type a inferences are correct, the distance-2 edge from node x to node z does not need to be created and thus we can save computation time. For type b and type c, we may add an edge from x to z and a *confidence factor* (see Subsection 3.1) to find potential errors that cannot be identified by transitivity.

A larger distance-d graph model can provide more information, but also needs more overhead to maintain it. For example, if we add a node and an edge to Figure 2, it produces many possible types among which only a few have meaningful inferences. In practice, the distance-2 graph model is sufficient to cover the information about pieces exchanges that are important in the transition from the middle game to the endgame.

3. RESOLVING CONFLICTS IN THE GRAPH MODEL

3.1 The Graph Model with Meta-knowledge

First, we introduce a *confidence factor* for an edge e, denoted by $CF(e)$, which is set by Chinese chess masters to indicate the confidence of the correctness in different types of inferences to find potential errors. Note that $CF(e)$ of any distance-1 edge is 100% because the edge applies the piece additive rule. There is a plethora of distance-2 edges of type b and type c, but many of them cannot infer meaningful results. Hence, we many apply some effective meta-knowledge to help the distance-2 graph model detect more potential errors. An example of distance-2 graph model is shown in Figure 3. The inference $G \rightarrow B$ is a type a inference because $G \rightarrow E$ and $E \rightarrow B$ can derive $G \rightarrow B$. Therefore, we do not need to add an edge to check the consistency of $G \rightarrow B$. The edges $B \rightarrow C$ and $F \rightarrow B$ are type b and c inferences respectively. For distance-2 edges incurred by type b and c, there are piece exchanges, and therefore both cannot be inferred using the piece additive rule. As an example, we use a meta-knowledge rule consisting of $CF(e) = 95\%$ and $CF(e) = 90\%$ for exchanging a strong piece with a defending piece and exchanging a strong piece with a pawn, respectively (see Figure 3).

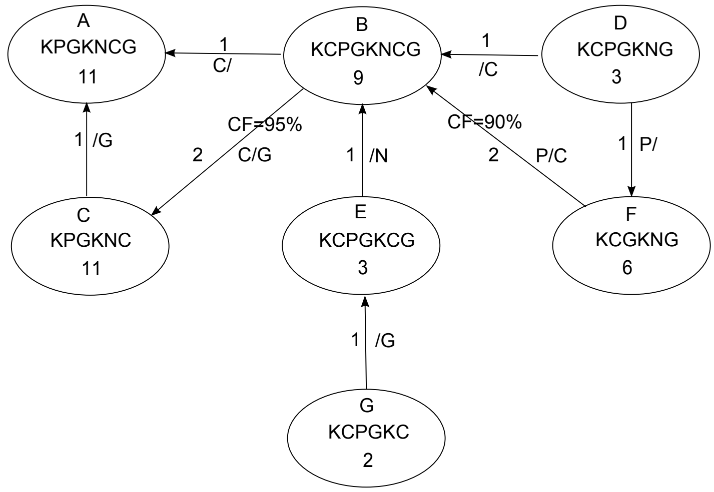


Figure 3: An example of the graph model with heuristics.

3.2 The Generalized Conflict Resolution Algorithm

As described in Definition 4, the distance-k graph model $\Pi_k = (\Gamma, D_k, \Phi)$ uses D_k to find the neighbors of the node x, and Φ to identify conflicts. Let e be a directed edge from x to y or from y to x. We define the *weighted neighbor sum* of a node x, denoted by $wns(x)$, to be $\sum_{\forall e \in \{\overline{xy}: y \in N_k(x)\}} CF(e)$. Let $F_\Phi(e)$ be 1 if Φ judges that a conflict occurs; and 0 otherwise. We also define the *weighted neighbor conflict sum* of x, denoted by $wncs(x)$, to be $\sum_{\forall e \in \{\overline{xy}: y \in N_k(x)\}} CF(e)F_\Phi(e)$. The larger $wncs(x)$ is, the more conflicts Φ detects. Then, the *weighted error ratio* of x, denoted by $wer(x)$, is evaluated by $wncs(x)/wns(x)$.

When a conflict occurs, we compute the total weighted neighbor conflict sum of Γ and choose to modify the node x of the maximum weighted error ratio. To modify x, we have to choose another advantage score of node x to reduce the overall conflicts in Γ. We define the *error score* as follows. First, the weighted error ratios are classified into ten levels, that is, level i represents the range of the weighted error ratio r by the class $10 * i\% \leq r < 10 * (i + 1)\%$, where $i = 0, ..., 8$, and the range of level 9 is $90\% \leq r \leq 100\%$. For example, in Figure 4(a), the weighted error ratios of nodes A and B are 88.7% and 1.1% respectively, and therefore their levels are 8 and 0 respectively. For each i, $0 \leq i \leq 9$, we count the number of nodes in Γ of which the weighed error ratios are within the range defined by level i, denoted by $wer\_level[i]$. Then, let the error score be $\sum_{i=0}^{9} 2^i * wer\_level[i]$. Note that if there is a node with a high weighted error ratio in Γ, then the error score will be very large.

Algorithm 1 Generalized conflict resolution algorithm

```
{* compute the weighted neighbor sum and the weighted neighbor conflict sum of each node *}
procedure ComputeWeightedSum(x, D_k, Φ)
    x.wncs = 0                                  {* x.wncs is used to store wncs(x) *}
    x.wns = 0                                   {* x.wns is used to store wns(x) *}
    for all y in N_k(x) do
        Let e be the edge between x and y       {* either x → y or x ← y *}
        x.wns = x.wns + CF(e)                   {* the weighted neighbor sum*}
        x.wncs = x.wncs + CF(e)F_Φ(e)           {* the weighted neighbor conflict sum *}
    end for
end procedure

{* compute and return weighted neighbor conflict sums of all nodes *}
function ConflictDiscovery(Γ, D_k, Φ) return the total weighted neighbor conflict sum
    total_wncs = 0
    for all x in Γ do
        ComputeWeightedSum(x, D_k, Φ)
        total_wncs = total_wncs + x.wncs
    end for
    return total_wncs
end function

{* update the weighted neighbor conflict sums of node x and its neighbors *}
procedure UpdateNodeConflict(x, D_k, Φ)
    ComputeWeightedSum(x, D_k, Φ)
    for all y in N_k(x) do
        ComputeWeightedSum(y, D_k, Φ)
    end for
end procedure

{* subtract old weighted neighbor conflict sums of node x and its neighbors *}
function SubtractWNCS(x, total_wncs, D_k, Φ) return a weighted neighbor conflict sum
    total_wncs = total_wncs − x.wncs
    for all y in N_k(x) do
        total_wncs = total_wncs − y.wncs
    end for
    return total_wncs
end function

{* add new weighted neighbor conflict sums of node x and its neighbors *}
function AddWNCS(x, total_wncs, D_k, Φ) return a weighted neighbor conflict sum
    total_wncs = total_wncs + x.wncs
    for all y in N_k(x) do
        total_wncs = total_wncs + y.wncs
    end for
    return total_wncs
end function

{* find the node of which the weighted error ratio is the maximum *}
function CandidateSelection(Γ) return a node
    candidate = null
    wer = 0
    for all x in Γ do
        if x.modified = false and x.invariable = false and x.wncs/x.wns > wer then
            candidate = x
            wer = x.wncs/x.wns
        end if
    end for
    return candidate
end function
```

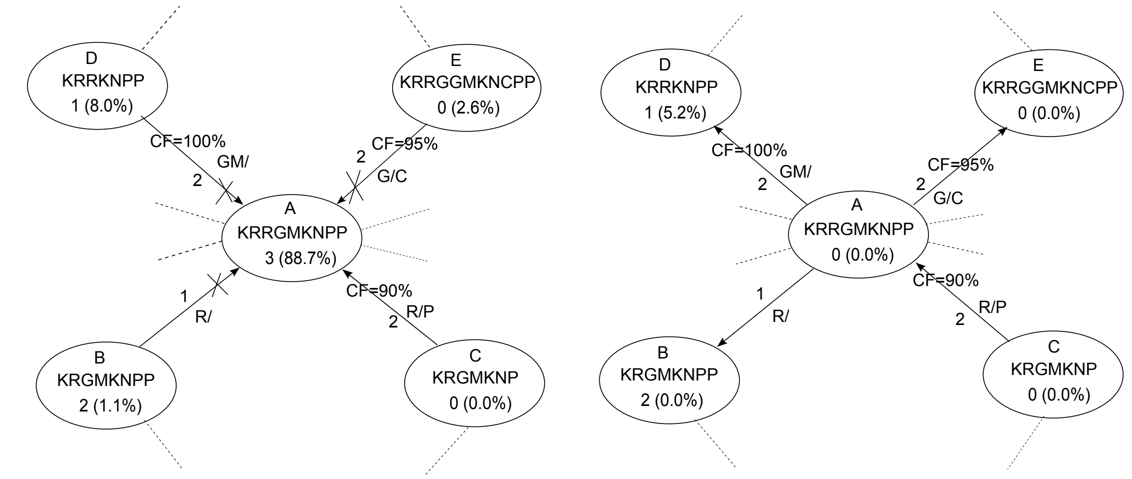
Algorithm 1 Generalized conflict resolution algorithm (cont.)

```
{* select an advantage score for node x so that the error score of Γ is the minimum *}
function ScoreValueSelection(x) return the best advantage score
    initialize wer_level[i] to be 0 for i = 0, ..., 9
    min_err_score = ∞
    for v = 0 to 11 do
        x.v = v
        UpdateNodeConflict(x, D_k, Φ)
        {* compute error score score for the advantage score v *}
        for all y ∈ Γ do
            get the level i of the weighted error ratio y.wncs/y.wns
            wer_level[i] = wer_level[i] + 1
        end for
        score = 0
        for i = 0 to 9 do
            score = score + 2^i * wer_level[i]
        end for
        if score < min_err_score then
            min_err_score = score
            best_v = v
        end if
    end for
    return best_v
end function

{* resolve conflicts in the graph Γ *}
procedure GeneralizedConflictResolution(Γ, D_k, Φ)
    total_wncs = ConflictDiscovery(Γ, D_k, Φ)
    org_wncs = ∞
    while total_wncs > 0 and total_wncs < org_wncs do
        org_wncs = total_wncs
        for all x ∈ Γ do
            {* set the modified flag as not modified *}
            x.modified = false
        end for
        while (x = CandidateSelection(Γ)) ≠ null do
            total_wncs = SubtractWNCS(x, total_wncs, D_k, Φ)
            x.v = ScoreValueSelection(x)
            UpdateNodeConflict(x, D_k, Φ)
            total_wncs = AddWNCS(x, total_wncs, D_k, Φ)
            x.modified = true
        end while
    end while
end procedure
```

The generalized conflict resolution algorithm is shown in Algorithm 1. In our knowledge base, the advantage scores of certain material combinations are labeled by Chinese chess masters as “invariable”. The nodes that represent these material combinations should not be modified by the algorithm.

In GeneralizedConflictResolution(), we compute the weighted neighbor conflict sums of all nodes in Γ and evaluate their summation, denoted by $total\_wncs$. Then, we iterate the following steps until $total\_wncs$ never decreases. In each iteration, we find the node x of which the weighted error ratio is the maximum. Next, the advantage score of x is modified to the one that generates the minimal error score of Γ. Finally, we compute $total\_wncs$ again. Note that each node is processed once in the inside loop.



(a) distance-1 graph model (b) distance-2 graph model

Figure 4: Modifying the material combination KRRGMKNPP. The edges marked by "X" are conflicting edges. The values in the nodes represent their advantage scores and weighted error ratios.

In the algorithm, the inside loop checks the whole graph Γ for modification; the outside loop continues when there is at least one change in the inside loop. To accelerate the conflict-detection algorithm, we compute and store the weighted neighbor sum and the weighted neighbor conflict sum for each node. Let N_k^* be the maximum size of distance-k neighbors for all nodes. CONFLICTDISCOVERY() obtains the weighted neighbor conflict sum of the whole graph. Its time complexity is O($N * N_k^*$). We apply a faster function UPDATENODECONFLICT() to check the consistency of only the node x and the neighbors of the node x inside the loop in the algorithm, which takes O(N_k^*) time. The time needed for the inner loop is $O(N + N_k^*)$. The outer loop may run at most N times. As a result, the time complexity of the algorithm is $O(N(N + N_k^*))$. Because N_k^* is much smaller than N, the algorithm can be viewed as $O(N^2)$.

Figure 4 is a real example in our knowledge base. In Figure 4(a), the distance-2 weighted neighbor conflict sum is 7381.1 and the error score is 396720 in Γ. The node A has 51 distance-2 edges. Its largest weighted error ratio is 88.7%, and its advantage score $A.v = 3$ that is wrong according to expert knowledge. Now, we call the conflict resolution algorithm. CANDIDATESELECTION() selects the node A to be modified because its weighted error ratio is the largest. Then, in SCOREVALUESELECTION(), we find the advantage score of 0 among $0, ..., 11$ resulting in the smallest error score, that is, 362400, and therefore the total weighted neighbor conflict sum becomes 7191.6. The algorithm sets $A.v = 0$, and as a result, it reduces the total weighted neighbor conflict sum of Γ the most under the condition that only one node is allowed to change its advantage score, as shown in Figure 4(b).

The original data is constructed by an automatic strategy (Chen *et al.*, 2008) that produces many unstable advantage scores. When there are conflict nodes in Γ, we call the conflict-resolution algorithm to reduce the number of conflicts. When the algorithm stops, the human expert does a small number of revisions and then calls the algorithm again. After a few iterations, we obtain a consistent knowledge base.

4. EXPERIMENTS AND DISCUSSIONS

The initial knowledge base was END6C from our previous work (see Chen *et al.*, 2008; Chen *et al.*, 2009), which is consistent and contained $69,595$ material combinations. END6C was further extended to a knowledge base with $123,985$ material combinations, which is called END12CR (Chen *et al.*, 2012). By adding 762 endgames with a high probability to be used in real games, the resulting knowledge base, called END12C+, contained $124,747$ material combinations. Subsequently, we constructed a larger knowledge base called END14CR that contained $140,320$ material combinations and used the distance-2 graph model with a set of meta-knowledge rules to resolve conflicts.

In this section, we will demonstrate that the END14CR knowledge base is the best version. In Section 4.1, we show the differences between the knowledge bases created by previous methods and that created by the distance-2 graph model. In order to show the improvement in playing strength, we perform self-play experiments. In Section 4.2, we propose a method to analyze the results of the self-play experiments. Then, in Section 4.3, we show the results of the self-play experiment.

4.1 The Difference Analysis

Because END14CR contains more material combinations than END6C and END12C+, we construct and compile two extra versions, END6C* and END12C+*, to ensure a fair comparison. The material combinations in END6C* and END12C+* have the same numbers of material combinations as those in END6C and END12C+ respectively, but their advantage scores are taken from END14CR. The experiment consists of two comparisons, each comparison is denoted by V_1 vs. V_2. Table 1(a) and (b) show the results of END6C vs. END6C* and END12C+ vs. END12C+* respectively. Given a 2-D table T, $T_{i,j}$ represents the cell at the intersection of row i and column j, meaning that the number of material combinations of which the advantage scores are i in V_1 and j in V_2. Let T_1 be the set of the elements in Table 1(a) and T_2 be those in Table 1(b). Let $n(T)$ be the total number of material combinations in a knowledge base in T. We find the following values $n(T_1) = 69,595$ and $n(T_2) = 124,747$. The sum of all non-diagonal values in T_1 and T_2 are $39,985$ and $36,111$ respectively. In order to compute the difference of the two knowledge bases, we compute the root mean square value of a table T as $RMS(T) = \sqrt{(\sum_{i=0}^{11}\sum_{j=0}^{11}(i-j)^2 * T_{i,j})/n}$. Because the value of $|i-j|$ is at most 11, we have $0 \leq RMS(T) \leq 11$. $RMS(T) = 0.0$ means that the two knowledge bases being compared in the 2-D table T are equal; $0.0 \leq RMS(T) \leq 1.0$ means that most entries in the two knowledge bases compared are no more than 1; $1.0 \leq RMS(T) \leq 2.0$ means that most entries in the two knowledge bases are different by at least 1 and at most 2.

According to our results, $RMS(T_1) = 1.37$ and $RMS(T_2) = 1.30$. This indicates that, in most cases, the difference of the advantage scores of a material combination before and after modification is at least 1.

4.2 Significance Analysis of Self-play Experiments

We use the multinomial distribution model to analyze the significance of an experimental result. Consider two copies of the same program playing against each other in a self-play experiment. In this case, the outcome of each game is an independent random trial that can be modeled as a trinomial random variable. Assume that for the copy playing first,

$$Pr(game_{first}) = \begin{cases} p & \text{if it won the game;} \\ q & \text{if the game was a draw;} \\ 1-p-q & \text{if it lost the game.} \end{cases}$$

Hence, for the copy playing second,

$$Pr(game_{second}) = \begin{cases} 1-p-q & \text{if it won the game;} \\ q & \text{it the game was a draw;} \\ p & \text{if it lost the game.} \end{cases}$$

Assume that the experiment comprises of $2n$ games: g_1, g_2, ..., g_{2n}. Let g_{2i-1} and g_{2i} be the ith pair of games, also called a *round*, where $i = 1, ..., n$. From the prospective of the the copy that plays g_{2i-1}, the outcome of the ith pair of games will be a random variable, denoted by X_i. Assume we assign the score x for a winning game, 0 for a draw, and $-x$ for a loss. The outcome of X_i and its occurrence probability is thus

$$Pr(X_i) = \begin{cases} p(1-p-q) & \text{if } X_i = 2x; \\ pq + (1-p-q)q & \text{if } X_i = x; \\ p^2 + (1-p-q)^2 + q^2 & \text{if } X_i = 0; \\ pq + (1-p-q)q & \text{if } X_i = -x; \\ (1-p-q)p & \text{if } X_i = -2x. \end{cases}$$

Table 1: Difference analysis of endgame knowledge bases

(a) END6C vs. END6C* (row: END6C, column: END6C*, $d_{rms} = 1.37$)

	0	1	2	3	4	5	6	7	8	9	10	11
0	6,024	291	99	62	19	0	6	2	0	4	0	2
1	10,250	5,452	1,277	569	49	38	2	7	13	0	0	0
2	272	811	1,630	1,112	463	339	59	127	271	30	0	0
3	223	300	405	755	1,250	159	472	251	178	68	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0
5	0	5	144	152	77	454	89	77	152	144	5	0
6	0	0	0	4	79	14	1,434	79	4	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	68	178	251	159	472	1,250	755	405	300	223
9	0	0	30	271	127	339	59	463	1,112	1,630	811	272
10	0	0	0	13	7	38	2	49	569	1,277	5,452	10,250
11	2	0	4	0	2	0	6	19	62	99	291	6,024

(b) END12C+ vs. END12C+* (row: END12C+, column: END12C+*, $d_{rms} = 1.30$)

	0	1	2	3	4	5	6	7	8	9	10	11
0	21,242	775	656	196	96	9	14	4	4	0	0	0
1	969	11,911	1,740	775	168	113	23	16	10	2	1	0
2	73	613	5,992	2,290	594	494	22	50	175	27	2	0
3	1	84	1,141	2,965	2,293	873	726	571	730	218	61	0
4	0	0	0	0	0	0	0	0	0	0	0	0
5	0	63	608	407	215	2,981	185	215	407	608	63	0
6	0	0	1	2	53	10	1,435	53	2	1	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0
8	0	61	218	730	571	873	726	2,293	2,965	1,141	84	1
9	0	2	27	175	50	494	22	594	2,290	5,992	613	73
10	0	1	2	10	16	113	23	168	775	1,740	11,911	969
11	0	0	0	4	4	9	14	96	196	656	775	21,242

The mean $E(X_i) = 0$. The standard deviation σ of X_i is $\sqrt{E(X_i^2)} = x\sqrt{2pq + (2q + 8p)(1 - p - q)}$, and X_i is a multinominally distributed random variable. Let $X[n] = \sum_{i=1}^{n} X_i$, and let s be the score we want to obtain.

After playing n rounds, the probability of getting a score s is $Pr(|X[n]| = s)$. The mean of $X[n] = 0$. The standard deviation of $X[n]$, σ_n, is $x\sqrt{n}\sqrt{2pq + (2q + 8p)(1 - p - q)}$. Assume that two programs play against each other and have obtained a score of s, $s > 0$. After playing n rounds, if $Pr(|X[n]| < s)$ is close to 1, the experiment result s can be regarded as statistically significance, that is, a program with a higher score is better than the opponent because the result only happens with a low random probability close to 0.

In Chinese chess, we set $x = 1$. By analyzing $63,548$ game records played among masters as collected in http://www.dpxq.com, we obtained $p = 0.3918$ and $q = 0.3161$; thus, $1 - p - q = 0.2921$. The standard deviation of $X[n]$, $\sigma_n \approx \sqrt{1.16n}$.

4.3 The Self-play Results

For the experiment, 680 positions that belong to the popular opening categories from our opening book are selected by Chinese chess masters. The first and second plies of the path from the beginning to the position are chosen from **"First Ply"** and **"Second Ply"** in Table 2 respectively. The following plies are chosen randomly to reach different positions. The experiment was performed on four servers, each of which had an AMD64 CPU (24 cores, 3.47GHz) and a 96GB memory. In each game, we utilized the timing scheme used in the Computer Olympiad, i.e., a player was allowed a total playing time of 30 minutes and lost the game if the allocated time was used. A round consists of two games with each player playing as the red side and the black side once. Each pair of versions that apply different knowledge bases was used to play 680 rounds.

Table 2: The opening categories. Each element is denoted by "ply taken (the number of such positions)". The notations are described in Yen *et al.* (2004).

First Ply	Second Ply
Ch2-e2 (400)	Hh9-g7 (240), Hb9-c7 (80), Ch7-e7 (80)
Pg3-g4 (200)	Ch7-g7 (120), Pc6-c5 (40), Hb9-c7 (40)
Eg0-e2 (80)	Ch7-d7 (40), Pg6-g5 (20), Hb9-c7 (20)

The results of the comparison of two versions, V_1 vs. V_2, are shown in Table 3, in which the first column means V_1, and the first row means V_2. The results are significant in statistical respect when the probability $\pi \geq 0.95$; and not significant otherwise. When the results are significant, V_1 is better if $s > 0$, and V_2 is better if $s < 0$.

Knowledge bases are used to guide the search to the right direction, that is, guiding the search to advantageous endgames or pruning the disadvantageous choices. Therefore, we analyze the results of the games that reach the endgame for which endgame knowledge bases are typically used. In our definition, a game reaches the endgame phase when each side has at most four strong pieces. Note that a rook is counted as two strong pieces. Each cell in Table 3 contains a triple (W, L, D) which are the numbers of wins, losses, and draws, respectively. With respect to V_1, we have the number of rounds n in which the endgame is reached, a score s, a probability $\pi = Pr(|X[n]| < s)$, and a standard deviation σ, as described in Subsection 4.2. A round consists of one pair of games in which two versions play red and black once. In Table 3, END6C* performs better than Trivial. Though from the experimental data, the program with END14CR performs better than the one with END12C+*, the results are not statistically significant. We conjecture that it may be the case that END14CR does not have a much larger number of practical endgames that can be used in tournaments than END12C+*. Both END12C+* and END14CR perform much better than Trivial and END6C. This clearly indicates that a knowledge base with a larger size performs better.

In summary, we demonstrated the utility of the proposed meta-knowledge rules. By using our system, it is feasible to construct an endgame knowledge base of a large magnitude which is impossible to construct manually. The larger the size of the knowledge base is, the more useful the knowledge base is in practice. However, when the endgame gets larger, it becomes easier to contain conflicts. Of course, we can use the proposed meta-knowledge rules to find these conflicts, and try to resolve them. Our conclusion therefore reads that the extended knowledge base performs better than all the previous versions.

Table 3: The endgame results of the self-play.

		Trivial	END6C*	END12C+*	END14CR
Trivial	W, L, D		57, 160, 467	77, 212, 415	77, 233, 408
	n, s, π, σ		342, -103, 1.000, 19.9	352, -135, 1.000, 20.2	359, -156, 1.000, 20.4
END6C*	W, L, D	160, 57, 467		149, 221, 382	121, 263, 320
	n, s, π, σ	342, 103, 1.000, 19.9		376, -72, 0.999, 20.9	352, -142, 1.000, 20.2
END12C+*	W, L, D	212, 77, 415	221, 149, 382		127, 149, 194
	n, s, π, σ	352, 135, 1.000, 20.2	376, 72, 0.999, 20.9		235, -22, 0.795, 16.5
END14CR	W, L, D	233, 77, 408	263, 121, 320	149, 127, 194	
	n, s, π, σ	359, 156, 1.000, 20.4	352, 142, 1.000, 20.2	235, 22, 0.795, 16.5	

5. CONCLUDING REMARKS

In Chinese chess, choosing the suitable endgame positions from a mid game in transition to an endgame is important. Therefore, endgame evaluation functions should have relevant information on what is an advantage in various types of endgames. Because the size of the information is large, we construct a knowledge base by an automatic extension algorithm for the problem at hand. However, there may arise conflicts among the generated knowledge items in the knowledge base. In this article, we propose a distance-k graph with a confidence factor to model an algorithm's ability to find conflicts in a graph. Our first concluding remark is that the distance-2 graph model, which considers particularly information on the piece exchanges, is an effective method for exploiting the endgame knowledge base.

The results of the self-play experiments show that the proposed distance-2 graph model can be used to construct and compile a high quality knowledge base. Our second concluding remark is that any search algorithm with an accurate knowledge base of larger size has a better performance in real games. Moreover, the more information of real games is in the knowledge base, the higher the probability is that knowledge is applied during the search.

The endgame knowledge bases described in this article provide an expandable carrier of knowledge about endgames. They are the transition carrier between mid-game and endgame. They are sufficient to lead the game to win positions, while actual endgame databases have the obligation to win, especially in many hard-to-win positions. Our third concluding remark is that the endgame knowledge bases and the endgame databases should be integrated in a Chinese chess program. Thus, the endgame knowledge bases should help the search algorithm to go into advantageous endgames. Subsequently, the endgame databases should play the precise moves to win in a certain endgame. Our fourth concluding remark is that the strategy of aggregating a large amount of knowledge with the help of a conflict resolution algorithm is a novel idea, viz. that a framework should be used to obtain knowledge. Moreover, the constructed endgame knowledge is excellent material for human players to learn about endgames. Finally, we remark that the endgame knowledge base can also be integrated into a tutoring system.

6. ACKNOWLEDGMENT

This research is supported in part by National Science Council of Taiwan under Grant 100-2221-E-001-011-MY3 and 102-2221-E-033-033-MY2. The authors would thank Wu-Chang Guo who is a Chinese chess 7-dan master for his help on expert knowledge.

7. REFERENCES

Chen, B. N., Chang, H.-J., Hsu, S. C., Chen, J.-C., and Hsu, T. s. (2014). Advanced meta-knowledge for Chinese chess endgame knowledge bases. *ICGA Journal.* (submitted).

Chen, B. N., Liu, P. F., Hsu, S. C., and Hsu, T. s. (2008). Knowledge inferencing on Chinese chess endgames. *Computers and Games*, Vol. LNCS 5131, pp. 180–191.

Chen, B. N., Liu, P. F., Hsu, S. C., and Hsu, T. s. (2009). Conflict resolution of Chinese chess endgame knowledge base. *Advances in Computer Games*, Vol. LNCS 6048, pp. 146–157.

Chen, B. N., Liu, P. F., Hsu, S. C., and Hsu, T. s. (2012). Aggregating consistent endgame knowledge in Chinese chess. *Knowledge-Based Systems*, Vol. 34, pp. 34–42.

Chen, J. C. and Hsu, S. C. (2001). Construction of online query system of opening database in computer Chinese chess. *The 11th Conference on Artificial Intelligence and Applications.*

Ciancarini, P. and Dalla Libera, F. (1997). Decision making under uncertainty: a rational approach to Kriegspiel. *Advances in Computer Chess 8*, pp. 277–298.

Ciancarini, P. and Gaspari, M. (1989). A knowledge-based system and a development interface for the middlegame in Chess. *Advances in Computer Chess 5*, pp. 219–230.

Herik, H. J. Van den, Uiterwijk, J. W. H. M., and Van Rijswijck, J. (2002). Games solved: now and in the future. *Artificial Intelligence*, Vol. 134, pp. 277–311.

Rubin, S., Lee, G., Sivaramakrishna, C., Pedrycz, W., and Chen, S. (2008). Learning conceptual Chess for testing evolutionary programming versus a reasoning-based soft expert system: the KASER. *Automation Congress*, pp. 1–6.

Schaeffer, J., Burch, N., Björnsson, Y., Kishimoto, A., Müller, M., Lake, R., Lu, P., and Sutphen, S. (2007). Checkers is solved. *Science*, Vol. 317, No. 5844, pp. 1518–1522.

Wu, P. S., Liu, P. Y., and Hsu, T. s. (2006). An external-memory retrograde analysis algorithm. *Computers and Games*, Vol. LNCS 3846, pp. 145–160.

NOTES

GARDNER'S MINICHESS VARIANT IS SOLVED

Mehdi Mhalla[1] *and Frédéric Prost*[2]

Université de Grenoble - LIG, B.P. 53 - 38041 Grenoble Cedex 09, France

ABSTRACT

A 5×5 board is the smallest board on which one can set up all types of chess pieces as a start position. We consider Gardner's minichess variant in which all pieces are set as in a standard chessboard (from Rook to King). This game has roughly 9×10^{18} legal positions and is comparable in this respect with checkers. We weakly solve this game: we prove its game-theoretic value and give a strategy to draw against best play for White and Black sides. Our approach requires surprisingly little computing power. We give a human readable proof. The way the result is obtained is generic and could be generalized to bigger chess settings or to other games.

1. INTRODUCTION

Solving popular games like Othello, Checkers or Chess is tantamount to finding the grail in the field of computer games. The resolution of checkers (Schaeffer *et al.*, 2007) put a mark in the field in the sense that the space search of this game is enormous (5×10^{20}) and the difficulty to make correct move decisions fairly high.

The game of chess has been recognized as the ultimate challenge in artificial intelligence for a long time. Since the early days of computer science, chess and computers have interacted together, see for example (Prost, 2012). Nowadays computers have superhuman strength and the game is partially solved: endgame databases up to few pieces have been computed. The most famous ones being the Nalimov tables (6 pieces). Recently Lomonosov endgame tablebases (Ltd., 2013) have been computed and give perfect play for 7 pieces (the size of the tablebase is 140 Terabytes). Nevertheless, the resolution of chess remains too difficult to be imagined: the number of legal positions is something around 10^{45} (Allis, 1994) and given a position, finding a good move (a move move leading to a losing position for the opponent if there is one, otherwise a move leading to a draw position, otherwise any move) is usually very difficult (the amount of chess literature is a proof by itself).

Some studies have been done to resolve particular cases of chess on very small boards. Notably, 3×3, 3×4 and 4×4 (the latter one being limited to 9 pieces on the board) chess variants have been solved by K. Kryukov (Kryukov, 2004; Kryukov, 2009; Kryukov, 2011). In these variants there is no starting position as in traditional chess. Positions are treated as puzzles. Each variant is strongly solved in the sense that the game-theoretic value of all legal positions is determined together with the perfect play associated. The number of legal positions is roughly 3×10^{15} for the 4×4 variant (Kryukov, 2011).

In this paper we study the variant called Gardner's Chess. Martin Gardner proposed Minichess in Scientific American (March 1962), later reprinted in (Gardner, 1989). It is played on a 5×5 board, the initial position is the initial position of chess but for the three pieces on the King side that are removed. The rules are the ones of classical chess without the two squares move for pawns, en passant captures and castling. This variant has roughly 9×10^{18} legal positions. This variant has been played extensively in Italy by correspondence (Pritchard, 2007). The results of finished games were the following:

[1]Mehdi.Mhalla@imag.fr
[2]Frederic.Prost@imag.fr

- White victory 40%
- draw 32%
- Black victory 28 %

However, in the late 90's the game was thought to be a draw. The text accompanying the implementation of Minichess in Zillions of games (Mallett and Lefler, 1998): ''This setup was adopted by the AISE (Associazione Italiana Scacchi Eterodossi) in 1978. Recent play has suggested the game is a draw with best play.''.

2. RESULTS

The game-theoretic value of Gardner's Chess is a draw. We prove this by giving two oracles, one for White and one for Black. Both oracles can force a draw versus best play. The intersection of the two oracles gives flawless games. Thus Gardner's chess is weakly solved.

Our engine has found that in Mallett Minichess, White can force checkmate in at most 25 moves. Mallett Minichess is played on a 5x5 board as well. The starting position for White is, from left to right: Rook, Knight, King, Queen and Knight whereas for Black it is Rook, Bishop, King, Queen and Bishop. We give a PGN file justifying this result on our webpage : [*"http://membres-lig.imag.fr/prost/MiniChessResolution/"*]. Interestingly, it looks like, though we do not have a complete formal proof of that, Mallett Minichess becomes a draw if it is White that has two Bishops and Black that has two Knights in the starting position. It copes well with the intuitive feeling that the relative strength of Knights vs. Bishops should be increased on smaller boards.

The proof of the draw for Gardner Minichess is surprisingly small and can be totally checked by a human, with the help of a computer within few hours. Oracles are given in appendix A for the White side and appendix B for the Black side. Due to the lack of space we only give the oracle for one of the seven legal White first move (the interested reader can found the totality of the oracles in the research report (Mhalla and Prost, 2013) or in our web site dedicated to minichess : [*"http://membres-lig.imag.fr/prost/MiniChessResolution/"*]). The number of positions examined in all the oracles combined amounts to a little more than 1500 positions. This is very small with relation to the game itself since it is smaller than the number of reachable positions after 4 plies (for standard chess there are 72 000 positions after the 4 plies).

From this point of view our result strongly differs from the resolution of checkers despite the fact that space search and difficulty of decision are of the same order of magnitude in both games. Indeed, the proof of (Schaeffer *et al.*, 2007) is not checkable by humans: it has required an enormous computing power (hundreds of computers in parallel over a decade). Most of our work was achieved with consumer-grade laptop computers. We have adapted the open source Stockfish chess engine (Romstad *et al.*, 2010) to play Gardner's Chess mainly by restricting the movements to the part of the board and changing the promotion ranks. Sources, executables for several environments and various files, including the oracles in PGN format as well as the list of the perfect openings for Gardner's Chess, can be found at the author's Minichess Resolution page: [*"http://membres-lig.imag.fr/prost/MiniChessResolution/"*].

The main line of oracles were computed in a semi-automated way: we were mainly following the most equalizing line. It turns out that most of the deviations from the main line can be quickly decided. This is mainly due to the fact that in Gardner's chess pawns are immediately exchanged or blocked. Moreover, pieces cannot develop naturally since almost all free squares are controlled by pawns. Also, the fact that promotion happens quickly leads to some very rapid checkmates that allow an efficient pruning of the game tree.

Using these Oracles, it is impossible to lose. Oracle for White (resp. for Black) does not examine alternative choices for White (resp. Black) decision nodes but indicates how to answer every possible Black (resp. White) ''reasonable'' move. Unreasonable moves, i.e. moves that obviously lead to a position where it is clear that Black (resp. White) cannot win, and is indeed a loss, can be dealt with our engine. We provide an upper bound of the number of moves required to checkmate (our engine does not necessarily give the exact distance to checkmate). Nevertheless, in these positions, from a human point of view, it is easy not to lose. Our claim is that a standard chess player equipped with our engine cannot lose in Gardner Minichess even versus the best play.

As a by-product of our study on Gardner's Chess the analysis of perfect openings shows the positions in which the evaluation of Stockfish is faulty. Indeed for some positions while showing largely "won" evaluation (up to

+6) the position is completely equal. What is interesting is that these evaluation bugs can be found on a 8x8 board as well. Thus the analysis of these positions may help to improve the evaluation of Stockfish for standard chess games. A complete description of the openings in Gardner Minichess as well as a sample of tricky draws and difficult checkmates can be found at [*"http://membres-lig.imag.fr/prost/MiniChessResolution/"*].

3. CONCLUSION

The game-theoretical value of Gardner Minichess has been proved to be a draw. The game-theoretical value of Mallett Minichess has been proved to be a loss for Black. The proofs were done in a semi-automated way in which humans were guiding the engine. The authors were 'pushing' lines for which it was thought that a forced checkmate could be computed and backtracked once leaves were showing a upper bound of the distance to checkmate. This meta-algorithm leads to a very asymmetric way of selecting moves. For instance, when a position is thought to be decidable as a White win, very little time is spent on White decision nodes (since we 'know' the game to be won more or less no matter what). The idea is that enormous time and energy can be saved when the game-theoretic value of a position, rather than the most precise move or the shortest path to checkmate, is looked for. Indeed, when a game is thought to be winning, e.g. for White, one has only to provide one forced line (even if it is not the 'best' one) and thus can avoid exhaustive search at White decision nodes. It can be seen as a form of meta-negascout (Fishburn, 1981): we forced the engine to go deep in the analysis tree for the winning side without analyzing too much. Nevertheless, it is very different in the sense that the process is very asymmetric and guided by the fact that the overall evaluation of the position is known.

Note that the fact that the game is weakly solved does not change the fact that it contains a lot of very nice variations and very tricky checkmates as one can see from the analysis of different gardner variations that can be found at [*"http://membres-lig.imag.fr/prost/MiniChessResolution/"*].

Our meta-procedure can be fully automated and tuned to some given degree of precision (basically what is the threshold after which a position is considered as decided). It can be used for example to confirm that the other 5x5 variants are also a draw (even if we exchange Knight and Bishop's positions for Black in Gardner's starting position, it appears to be a draw). For future works we plan to implement it, test it, and extend it to be able to cope with larger chess variants to compute their game-theoretic values. A natural candidate would be Los Alamos chess which is played on a 6x6 board. Other games could also be considered.

Acknowledgments

We thank Francois Challier and Philippe Virouleau for their technical and technological help. We also thank anonymous referees for their constructive feedbacks and the time spent to thoroughly check the oracles.

4. REFERENCES

Allis, V. (1994). *Searching for Solutions in Games and Artificial Intelligence.* Ph.D. thesis, University of Limburg, Maastricht, The Netherlands.

Fishburn, J. P. (1981). *Analysis of Speedup in Distributed Algorithms.* Ph.D. thesis, University of Winconsin, Madison.

Gardner, M. (1989). *The Unexpected Hanging and Other Mathematical Diversions.* University of Chicago Press.

Kryukov, K. (2004). 3×3 chess. website : http://kirr.homeunix.org/chess/3x3-chess/.

Kryukov, K. (2009). 3×4 chess. website : http://kirr.homeunix.org/chess/3x4-chess/.

Kryukov, K. (2011). 4×4 chess. website : http://kirr.homeunix.org/chess/4x4-chess/.

Ltd., C. (2013). Lomonosov Endgame Tablebases. website : http://chessok.com/?page_id=27966.

Mallett, J. and Lefler, M. (1998). Zillion of Games 1. http://www.zillions-of-games.com/.

Mhalla, M. and Prost, F. (2013). Gardner's Minichess Variant is solved. *CoRR*, Vol. abs/1307.7118. http://arxiv.org/abs/1307.7118.

Pritchard, D. B. (2007). *The Classified Encyclopedia of Chess Variants.* J. Beasley.

Prost, F. (2012). On the impact of information technologies on society: an historical perspective through the game of chess. *Turing-100* (ed. A. Voronkov), Vol. 10 of *EPIC*, pp. 268–277, EasyChair.

Romstad, T., Costalba, M., Kiiski, J., Yang, D., Spitaleri, S., and Ablett, J. (2010). Stockfish. web site: http://stockfishchess.org/.

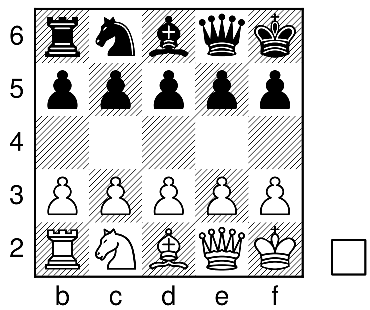
Schaeffer, J., Burch, N., Bjornsson, Y., Kishimoto, A., Muller, M., Lake, R., Lu, P., and Sutphen, S. (2007). Checkers is solved. *Nature*, Vol. 317, No. 5844, pp. 1518–1522.

5. APPENDICES

APPENDIX A: GARDNER : ORACLE FOR WHITE DRAW

We give an oracle for the White side of the Gardner variation. The objective is to force a draw versus the best play. Therefore, we give it as a tree of variations that needs no explanations on White nodes: it is maybe possible to find a shorter draw (or even win) but our aim is to have an oracle the most readable from a human point of view: the definitive judgment on the leafs of this tree are clear.

Since there are no choices to be explored for White nodes we adopt the following convention to name sub-variations: first we note the depth in the oracle, then we enumerate deviations from the main line by enumerating Black ¡moves from left to right, pawns come first, after we enumerate moves of the pieces following the lexical order going from left to right and top to bottom. Thus the variation $[3|1.3.7]$ is the one obtained by following the oracle until depth 3 and selecting as sub-variation move 1 as the first move for Black, then move 3 as second move for Black and 7 as the third move. We write $+-$ (resp. $-+$) when it is obvious that Black (resp. White) cannot win. We write $\sharp x \sharp_\bullet$ (resp. $\sharp x \sharp_\circ$) when there exists a forced checkmate of the Black King (res. White King) in x moves (though it is possible that shorter checkmates exist). Very often positions that look lost (because one side has a piece advantage for instance) can be fully decided by our engine as forced checkmates. Justifying lines are written like this: $\triangle$ **1 b4 c×b4 2 c×b4 d4**. Finally, the coordinate of the lower left square is **b2**. Hence the starting position is:



In this position the Black move identified by 1 is . . . b4 and move number 6 is . . . ♘b4, move number 7 is . . . ♘c4.

The non standard choice of b2 as coordinate of the lower left square comes from our adaptation of the Stockfish engine for Gardner minichess. Indeed, by placing pieces and pawns tihs way we avoid problems with en passant capture, and castling, for free.

We give the White oracle as a variation tree. After each move of the oracle we start by giving all lines in which a forced checkmate can be found using our engine.

1 b4

(**1…d4 2 b×c5** ♯47♯• △ after **2…♗×c5 3 f4** both the pressure on the b-file and on the b2-f6 diagonal are too strong to be sustained by Black. **1…e4 2 b×c5** ♯28♯•, the point is that on **2…♗×c5 3 d4** the threat of ♖×b5 combined with the lack of space for Black is too hard to be met. Other moves just lose a piece at least. **1…f4 2 b×c5** ♯24♯• △ 2... Bxc5 3.d4 the threat of ♖×b5. **1…♘×b4 2 c×b4** ♯24♯• White is a piece up for nothing. **1…♘d4 2 b×c5** ♯17♯• White is a piece up for nothing.):

- [1|1] **1…c4 2 d4** (**2…♗×b4 3 d×e5+** ♯29♯•, **2…♗c5 3 b×c5** ♯8♯•, **2…♘×d4** ♯29♯•, **2…f4 3 e4** ♯38♯• actually Black can only play one line, otherwise if he captures both the e and d-pawns he will soon lose material due to the multiple forks and lack of space △ **3…e×d4 4 ♘×d4 ♕e5 5 ♘×c6 ♖×c6 6 ♗×f4 ♕×f4** ,6…♕×c3 7 ♖c2 ♕d4+ 8 ♗e3 ♕e5 9 ♕d2 and the threat of ♗d4 is too strong, **7 e×d5** and Black cannot deal with the simultaneous threat on his Rook and of checkmate on e6.).
  - [1|1.1] **2…e×d4 3 e×d4** (**3…♗c5** ♯6♯•, **3…♘e5** ♯14♯•, **3…♕e3+** ♯6♯•, **3…♕e4** ♯10♯•, **3…♕e5** ♯12♯•, **3…♗e5**♯31♯•):
    - [1|1.1.1] **3…f4 4 ♕×e6+ ♔×e6** = White just has to move his King on e2-f2 and Black cannot break through. No matter what is the relative position of the two Kings, if the Black Knight takes on d4 or b4 White takes back with the Knight and the position is still blocked for Black and if Black plays …♗×b4 the position is ♯17♯• when kings on e-file and ♯24♯• when kings are in f-file. Finally, if Black plays …♘e5 white just takes it with d×e5 and if Black plays …♗e5 White just continue to move his king.
    - [1|1.1.2] **3…♕×e2+ 4 ♔×e2** = for the same reason as line [1|1.1.1].
  - [1|1.2] **2…e4 3 f4** = Black is in zugzwang and must give a piece. Due to the blocked nature of the position he can do it without losing but he cannot break through e.g. **3…♗e5 4 f×e5+ ♘×e5 5 d×e5+ ♕×e5 6 ♘d4** and White can simply moves back and forth with the Knight.
  - [1|1.3] **2…♘×b4 3 d×e5+** (**3…♗×e5 4 ♖×b4** ♯20♯•, **3…♔×e5 4 c×b4** ♯13♯•) +− :
    - [1|1.3.1] **3…♕×e5 4 ♘×b4** (**4…♗×b4 5 ♖×b4** ♯25♯•, **4…♕e6 5 e4** ♯17♯•, the threat of …♘×c5+ cannot be met efficiently △ **5…♗×b4 6 ♖×b4** and the extra Bishop tells with ♗e3 to follow, **4…♕d4 5 e×d4** ♯3♯•, **4…♕e4 5 f×e4** ♯9♯•, **4…♕f4 5 e×f4** ♯4♯•, **4…♕×d3 5 e×f4** ♯4♯•, **4…♕×b3 5 ♗×b3+** ♯5♯•, **4…d4 5 c×d4** ♯10♯•, **4…f4 5 e×f4** ♯10♯•, **4…♖c6 5 ♘×c6** ♯10♯•):
      - [1|1.3.1.1] **4…♗c5 5 ♘c2** = White blocks the position on the dark squares with ♘d4 and ♖b4 (and moves his Rook between b2-b4 if Black does not move. △ **5…f4 6 ♘d4 b4** (other moves leads to a loss for Black) **7 e×f4 ♗×d4+ 8 c×d4 ♕×e2+** (other moves lead to direct checkmate).
      - [1|1.3.1.2] **4…♔e6 5 ♘c2** +− similar to line [1|1.3.1.1].

1…c×b4 2 c×b4 All Knight and Bishop moves lose a piece and end up in a position where clearly Black cannot win (**2…♘d4** ♯23♯•, **2…♘×b4** ♯18♯•, **2…♗×b4** ♯15♯•, **2…e4 3 ♗c3+** ♯15♯• △ **3…♗e5 4 d4 ♗d6 5 f4** zugzwang : Black loses a piece on the next move))

- [2|1] we give here a suboptimal, but very easy to analyze, line for White since it is possible to prove a forced checkmate (but it is very hard to understand from a human point of view). **2…f4 3 e×f4 e×f4 4 ♕×e6+ ♔×e6 5 ♗c3**
  - [2|1.1] **5…d4 6 ♘×d4+ ♘×d4 7 ♗×d4 ♖c6 8 ♗c5** = Black cannot avoid the exchange of Bishops. It leads to a completely drawn endgame.
  - [2|1.2] **5…♗e5 6 d4 ♗d6** = see variation [2|1.3].
  - [2|1.3] **5…♔f5 6 d4** = Black can only moves his King and cannot untangle.

2…d4 3 e4 (**3…f×e4 4 f×e4** ♯8♯• since the threat of ♕f3+ is too strong, **3…♘×b4** ♯15♯•, **3…♗×b4** ♯15♯•, **3…♕b3** ♯12♯•, **3…♕c4** ♯8♯•, **3…♕c6**♯8♯•)

3…f4 4 ♗×f4 (All Black's alternatives lead to forced checkmate since they lose a piece for nothing **4…♗c5** ♯8♯•, **4…♕b3** ♯11♯•, **4…♕c4** ♯8♯•, **4…♕d5** ♯5♯•, **4…♕f5** ♯8♯•, **4…♘×b4** ♯25♯•, **4…♗×b4** ♯25♯•)

4…e×f4 5 ♕d2 (**5…♘e5 6 ♕×f4+** ♯2♯•, **5…♘×b4 6 ♘×b4** +− d5 cannot be protected and the threat of ♕×f4+ forbids **6…♗×b4**)

5...♝e5 6 ♔e2 = White just moves his King on e2-f2 and Black cannot untangle by ...♘e5 because of ♕×f4 and must otherwise give up a piece and cannot win.

APPENDIX B: GARDNER : ORACLES FOR BLACK DRAW

We now give an oracle from the Black point of view that forces the draw. So here we give no explanations for Black decision nodes and we explore all moves, as explained earlier, at White decision nodes. The interested reader may find all Black oracles, for all seven valid White first move, in (Mhalla and Prost, 2013).

APPENDIX B.1: White moves d4

1 d4 e4 (**2 ♘b4** ♯21♯◦, **2 ♕×b5** ♯12♯◦, **2 ♕c4** ♯9♯◦, **2 ♕d3** ♯10♯◦, **2 f×e4** ♯13♯◦)

- [1|1] **2 b4 c4** (**3 f×e4** ♯9♯◦, **3 ♖b3** ♯8♯◦, **3 ♕×c4**♯16♯◦, **3 ♕d3** ♯9♯◦) **3 f4 ♝×b4** (**4 ♖b3** ♯8♯◦, **4 ♖×b4 ♘×b4** ♯21♯◦ White is completely blocked and is soon forced to give up the Queen △ **5 ♘×b4 ♕d6 6 ♘c2 ♔e6 7 ♘b4 ♖c6 8 ♘c2 b4 9 c×b4 c3** , **4 ♕×c4** ♯15♯◦, **4 ♕d3** ♯10♯◦, **4 ♕f3** ♯11♯◦)
  - [1|1.1]**4 c×b4 ♛d6** = despite his extra piece White cannot win since he is blocked by his own pawns on dark squares.
  - [1|1.2]**4 ♘×b4 ♘×b4** = △ **5 ♖×b4 ♛d6** and White may only move his Rook, on **5 c×b4 ♛d6** is similar to line [1|1.1].
- [1|2] **2 c4 b×c4** (**3 b4** ♯15♯◦, **3 b×c4** ♯19♯◦, **3 ♘b4** ♯14♯◦, **3 ♗b4** ♯10♯◦, **3 f×e4** ♯10♯◦, **3 f4** ♯17♯◦, **3 ♕d3** ♯8♯◦, **3 ♕×c4** ♯12♯◦)
  - [1|2.1] **3 d×c5 ♝×c5** (**4 b4** ♯20♯◦ the Rook is soon to be lost, or material loss is even worse △ **4...e×f3 5 ♕×f3 ♕e5 6 ♘d4 ♗×d4 7 ♖c2 c3**, **4 ♘b4** ♯12♯◦, **4 ♘d4** ♯15♯◦, **4 ♗b4** ♯14♯◦, **4 ♕d3** ♯8♯◦, **4 ♕×c4** ♯11♯◦, **4 f4** ♯12♯◦ △ **4...♕d6** and White cannot avoid considerable material loss leading to forced checkmate.)

    4 ♝c3+ ♛e5 (**5 ♗×e5+** ♯19♯◦ the point is that after the recapture with the Knight on e5 White has no satisfactory way to keep his material advantage the weaknesses of d3 and b3 being keys △ **5...♘×e5 6 ♕d2 ♖×b3 7 ♘d4 ♘d3+ 8 ♔e2 ♖×b2 9 ♕×b2 ♘×b2**, **5 b×c4** ♯16♯◦, **5 f×e4** ♯13♯◦, **5 b4** ♯9♯◦, **5 ♘d4** ♯12♯◦, **5 ♘b4** ♯12♯◦, **5 f4** ♯7♯◦, **5 ♗d4** ♯9♯◦, **5 ♕×c4** ♯9♯◦, **5 ♕c3** ♯7♯◦)

    5 ♛d2 f4 the next White move is forced due to the threat on e3 (**6 ♔e2** ♯12♯◦, **6 f×e4** ♯13♯◦, **6 b4** ♯17♯◦, **6 ♘d4** ♯10♯◦, **6 ♗d4** ♯8♯◦, **6 b×c4** ♯8♯◦, **6 ♘b4** ♯8♯◦, **6 ♕d4** ♯7♯◦, **6 ♗b4** ♯7♯◦, **6 ♕e2** ♯6♯◦, **6 ♕×d5** ♯6♯◦, **6 ♕d3** ♯5♯◦)

    6 ♝×e5+ ♘×e5 here again the weaknesses of b3, d3 and e3 leave no choice for White (**7 ♘d4** ♯16♯◦, **7 ♕c3**♯12♯◦, **7 f×e4** ♯20♯◦, **7 ♕d4** ♯11♯◦, **7 ♔e2** ♯11♯◦, **7 ♘b4** ♯10♯◦, **7 ♕×d5** ♯8♯◦, **7 ♕b4** ♯6♯◦, **7 b×c4** ♯5♯◦, **7 ♕e2** ♯5♯◦, **7 ♕d3** ♯4♯◦)

    7 b4 f×e3+ (**8 ♕×e3** ♯14♯◦, **8 ♔e2 e×f3** checkmate)

    8 ♘×e3 ♘d3+ (**9 ♕×d3** ♯6♯◦)

    9 ♔e2 ♘f4+ 10 ♔f2 ♘d4+ = draw by repetition.
  - [1|2.2] **3 ♝c3 ♖×b3**

    4 ♖×b3 c×b3 (**5 f×e4** ♯9♯◦, **5 f4** ♯15♯◦, **5 ♘b4** ♯11♯◦, **5 ♗b4** ♯9♯◦, **5 ♗d2** ♯10♯◦, **5 ♕b5** ♯6♯◦, **5 ♕×c4** ♯6♯◦, **5 ♕d3** ♯5♯◦, **5 ♕d2** ♯15♯◦ by playing **5...f4** Black puts White in zugzwang)

    5 d×c5+ ♝e5 (**6 f×e4** ♯8♯◦, **6 f4** ♯6♯◦, **6 ♘b4** ♯7♯◦, **6 ♗b2** ♯8♯◦, **6 ♗b4** ♯9♯◦, **6 ♗d2** ♯7♯◦, **6 ♗d4** ♯9♯◦, **6 ♕d2** ♯16♯◦, **6 ♕d3** ♯4♯◦, **6 ♕c4** ♯6♯◦, **6 ♕b5** ♯6♯◦, **6 ♗×e5+** ♯13♯◦)

    6 ♘d4 ♝×d4 (**7 f×e4** ♯7♯◦, **7 f4** ♯6♯◦, **7 ♗b2** ♯11♯◦, **7 ♗b4** ♯5♯◦, **7 ♗d2** ♯6♯◦, **7 ♕d2** ♯11♯◦, **7 ♕d3** ♯3♯◦, **7 ♕c4** ♯8♯◦, **7 ♕b5** ♯10♯◦, **7 ♕b2** ♯11♯◦, **7 ♕c2** ♯2♯◦)

    - [1|2.2.1] **7 ♝×d4+ ♘×d4** (**8 c×d4** ♯3♯◦, **8 ♕d2**♯11♯◦, **8 f×e4**♯6♯◦, **8 c6♕**♯5♯◦ (other promotions as well), **8 f4**♯3♯◦, **8 ♕c2 b×c2♕+** checkmate)

    8 ♛b2 f4 (**9 ♕×d4+**♯17♯◦, **9 c6♕**♯11♯◦ (other promotions as well), **9 f×e4**♯7♯◦, **9 ♕d2**♯6♯◦, **9 ♕c2 b×c2♕+** checkmate)

9 e×d4 e×f3 (**10 ♕d2**♯3♯∘, **10 c6♗ ♕e3** checkmate, **10 c6♘ ♕e3** checkmate, **10 ♕c2 ♕e3** checkmate, **10 ♕e2 f×e2♕+** checkmate, **10 ♕×b3 ♕e2** checkmate, **10 ♕c3 ♕e2** checkmate)

10 c6♕ ♕×c6, promotion to Rook is handled similarly, (**11 ♔×f3**♯7♯∘, **11 ♕c3**♯4♯∘, **11 ♕d2**♯4♯∘, **11 ♕c2**♯3♯∘, **11 ♕e2**♯3♯∘)

11 ♕×b3 ♕e6 = Black will play . . . ♕e2+ and after Queen exchange the pawn endgame is drawn.

- [1|2.2.2] **7 e×d4 e3+ 8 ♕×e3 ♕×e3+ 9 ♔×e3** = the Black King just moves to e6-f6 and White King cannot break through. If the White Bishop goes to e5 either Black can play f4 and get room for his King or it means that White played f4 hence after . . . ♘b4 the Knight cannot be taken without stalemating the Black King.

– [1|3] **2 d×c5 ♗×c5** (**3 c4 b4**♯23♯∘, **3 f×e4** ♯14♯∘, **3 f4 ♕d6**♯14♯∘, **3 ♘b4 f4**♯16♯∘, **3 ♕×b5** ♯11♯∘, **3 ♕c4** ♯9♯∘, **3 ♕d3** ♯9♯∘, **3 b4** ♯16♯∘)

3 ♘d4 ♘×d4 (**4 b4** ♯12♯∘, **4 c4** ♯9♯∘, **4 f×e4** ♯10♯∘, **4 f4** ♯9♯∘, **4 ♖c2** ♯9♯∘, **4 ♕×b5** ♯8♯∘, **4 ♕c4** ♯8♯∘, **4 ♕d3** ♯7♯∘)

∗ [1|3.1] **4 c×d4 e×f3** (**5 b4** ♯12♯∘, **5 d×c5** ♯14♯∘, **5 e4** ♯5♯∘, **5 ♖c2** ♯10♯∘, **5 ♗c3** ♯10♯∘, **5 ♗b4** ♯8♯∘, **5 ♕c4** ♯12♯∘, **5 ♕×b5** ♯16♯∘)

- [1|3.1.1] **5 ♕d3 ♗d6** = on any reasonable move (**6 ♕e4** ♯6♯∘, **6 ♕c2** ♯32♯∘, **6 b4** ♯34♯∘, **6 ♗c3** ♯24♯∘, **6 ♗b4** ♯15♯∘, **6 ♕c3** ♯24♯∘, **6 e4** ♯12♯∘, **6 ♕×b5** ♯8♯∘, **6 ♕e2** ♯8♯∘, **6 ♕c4** ♯7♯∘, **6 ♕×f5+** ♯6♯∘) Black plays . . . ♕e4 and locks the position as in variation [1|3.1.3.1].

- [1|3.1.2] **5 ♕×f3 ♗d6** = on any reasonable move (**6 ♕×f5+** ♯7♯∘, **6 ♕f4** ♯8♯∘, **6 ♕×d5** ♯9♯∘, **6 ♕e4** ♯4♯∘ **6 e4** ♯37♯∘, **6 ♗b4** ♯22♯∘) Black plays . . . ♕e4 and locks the position as in variation [1|3.2.3.1].

- [1|3.1.3] **5 ♔×f3 ♕e4+ 6 ♔f2 ♗d6** (**7 ♖c2** ♯10♯∘, **7 ♗b4** ♯12♯∘, **7 ♕d3** ♯7♯∘, **7 ♕c4** ♯6♯∘, **7 ♕×b5** ♯8♯∘)

* [1|3.1.3.1]**7 b4 ♔e6** = Black just moves his King on e6-f6 and the position is blocked on the dark squares △ **8 ♕f3 ♔e6 9 ♕×e4+ f×e4**

* [1|3.1.3.2]**7 ♗c3 ♔e6** = since Black can move his King on e6-f6. If White takes on e4 then Black let the position closed with . . . f×e4 and the position is blocked.

* [1|3.1.3.3]**7 ♕f3 ♔e6** = indeed Black may only move his King around squares e6-f6, if White plays b4 then see line [1|3.1.3.1] and on ♗c3 see line [1|3.1.3.2].

∗ [1|3.2] **4 e×d4 ♗d6** (**5 ♖c2 f4** ♯23♯∘, **5 b4 f4** ♯18♯∘, **5 ♗e3 f4** ♯22♯∘, **5 ♕e3 f4** ♯22♯∘, **5 c4 b×c4** ♯23♯∘, **5 ♔e3 f4+** ♯23♯∘, **5 ♗f4** ♯11♯∘, **5 ♕×b5** ♯10♯∘, **5 ♕c4** ♯8♯∘, **5 ♕d3** ♯4♯∘, **5 ♗f4** ♯11♯∘)

- [1|3.2.1] **5 f×e4 ♕×e4** = if White exchanges Queen on e4 then with . . . f×e4 Black closes the position and with . . . ♖c6 White cannot do anything. If White does not exchange Queens then Black may just play his King (on **6 b4 f4** is ♯28♯∘).

- [1|3.2.2] **5 f4 e3+** = since Black follows with . . . ♕e4 and blocks the position.

2 f4 c4 (**3 ♕d3** ♯12♯∘, **3 ♕×c4 b×c4** ♯15♯∘, **3 ♕f3** ♯8♯∘)

– [2|1] **3 b×c4 d×c4** (**4 d5 ♕×d5** ♯22♯∘, **4 ♖×b5 ♖×b5** ♯10♯∘, **4 ♖b3** ♯9♯∘, **4 ♕×c4** ♯8♯∘, **4 ♕d3** ♯9♯∘, **4 ♕f3** ♯11♯∘)

∗ [2|1.1] **4 ♘b4 ♘×b4 5 c×b4 ♕d5** = the position is totally blocked on dark squares and Black may just moves his King on e6-f6.

∗ [2|1.2] **4 ♖b4 ♘×b4** (**5 d5** ♯8♯∘, **5 ♘×b4** ♯23♯∘, **5 ♕×c4** ♯5♯∘, **5 ♕d3** ♯7♯∘, **5 ♕f3** ♯6♯∘) **5 c×b4 ♕d5** = because the position is totally blocked and Black just moves his King to e6 f6. The only way to untangle for White is to sacrifice the Queen on c4 which lead to quick checkmate.

– [2|2] **3 ♘b4 ♘×b4** (**4 ♖c2** ♯8♯∘, **4 ♕×c4** ♯9♯∘, **4 ♕d3** ♯7♯∘, **4 b×c4** ♯15♯∘, **4 ♕f3** ♯7♯∘) = The draw is tricky to understand at first sight but becomes clear with the following variation **4 c×b4 ♗×f4** (**5 ♗c3** ♯16♯∘, **5 ♖c2** ♯15♯∘, **5 ♕d3** ♯6♯∘, **5 ♕×c4** ♯9♯∘, **5 ♕f3** ♯7♯∘). From here the idea is to build a blockade on dark squares.

- After **5 e×f4 ♖c6** (in order to be able to take with the Rook in the case of b×c4) = The blockade has been achieved and Black just moves his Queen on d6 and his King on e6 f6.

- **5 b×c4 b×c4 6 e×f4 ♖b5** = another blockade is built on dark squares and White cannot break through.

3 b4 ♗×b4 = due to the blocked position White cannot achieve anything, this type of position has already been treated in line [1|1] of this oracle for instance.

SELECTED ALGORITHMS FROM THE 2013 TOADS-AND-FROGS BLITZ TOURNAMENT

Wojciech Wieczorek[1] *Rafał Skinderowicz*[1] *Jan Kozak*[1] *Przemysław Juszczuk*[1]
Arkadiusz Nowakowski[1]

Sosnowiec, Poland

ABSTRACT

When a game is complex, analysis of the game state and finding a good move can often be time consuming. This note provides a description of four algorithms—two classic and two unconventional ones—for playing two-player games under severe time constraints. Toads-and-Frogs has been chosen as a testbed because, despite the simple rule set, the evaluation of its positions is an NP-hard problem. These algorithms were used in programs which took leading places in a computer Toads-and-Frogs blitz tournament held at the Silesian University in Poland.

1. INTRODUCTION

It happens pretty often that during a game one or both players (programs) get into serious time trouble. One of the reasons is a very complicated position and choosing a move needs to be preceded by long computations. The purpose of the present proposal is threefold. The first objective is to formulate alternative methods to play fast. This includes the classic alpha-beta game tree search, Monte-Carlo tree search (MCTS) with the upper confidence bound applied to trees (UCT), MCTS with fixed data storage, and the use of patterns and rules. The second objective is to determine the strength of these methods on the basis of a round-robin competition. The third objective is to report the results of a computer tournament where our four general-purpose algorithms competed with many specific algorithms dedicated solely to the Toads-and-Frogs game.

The present note's content is structured as follows. Section 2 explains the game. Section 3 describes our search approaches. Section 4 gives information on a computer Toads-and-Frogs tournament held on June 17, 2013. This section also presents a comparison between the performances of the four algorithms.

2. TOADS-AND-FROGS

Toads-and-Frogs is an abstract, two-player, strategy game invented by Richard Kenneth Guy. It is played on a $1 \times n$ strip of squares (a board), initially filled in a certain pattern with toads and frogs pieces with a number of unoccupied places. Toads-and-Frogs players take turns; on his[2] turn, a player may either move one square or jump over an opposing piece onto an empty square. Toads move only eastward, frogs only to the west. Naturally, from the rightmost square toads cannot move, neither can frogs move from the leftmost square. The first player who is unable to move loses.

Figure 1: An exemplary Toads-and-Frogs position.

[1]Institute of Computer Science, University of Silesia, Poland, corresponding author: email: wojciech.wieczorek@us.edu.pl

[2]For brevity, we use 'he' and 'his' whenever 'he or she' and 'his or her' are meant.

Figure 1 illustrates one of the possible positions in the game. In order to simplify the figures further (a notation), throughout the rest of the paper, we will represent toads by T, frogs by F, and empty squares by the symbol □. Thus the position from Figure 1 can be written simply as T□TF□FF. Although the game may start at any configuration, it is customary to begin with toads occupying consecutive squares on the leftmost end and frogs occupying consecutive squares on the rightmost end of the strip.

Determining the value of an arbitrary Toads-and-Frogs position is NP-hard, which was proved by Jesse Hull in 2000 (as mentioned by Thanatipanonda, 2008). A number of papers have been devoted to evaluating some particular positions of the game (e.g., Erickson, 1996; Thanatipanonda, 2008). This game has received so much coverage because of the simplicity and elegance of its rules and the beauty of its analysis.

3. DESCRIPTION OF ALGORITHMS

3.1 Alpha-beta game tree search

The presented algorithm is based on the classic alpha-beta game tree search. Specifically, the *iterative deepening* version was chosen, which allows for more efficient use of the available time (Marsland, 1986). In the presented work, an evaluation algorithm based on a few simple observations was used.

Note that for each sequence TTFF placed on consecutive squares of the board, no valid moves can be made as the player's piece may jump over at most one of the opponent's pieces. Such a sequence will be called a *hard lock*. Each hard lock splits the board into two independent *blocks*[3] consisting of the squares to the left and to the right of the hard lock, respectively. The pieces from either side cannot cross the hard lock. In the presented algorithm, the evaluation of the whole board is equal to the sum of evaluations for all the blocks.

It should be noted though that a player can easily block his opponent's moves if he puts two pieces on the consecutive squares, i.e., FF (TT in the case of the T player). Such a sequence will be called a *soft lock*. The soft lock, obviously, immobilizes the two pieces involved and hence reduces the number of available moves but can also significantly reduce the number of the opponent's moves. For this reason, in the algorithm all the blocks created by soft locks (FF and TT) are also evaluated.

The pseudocode of the board evaluation algorithm for the F player is shown as Procedure Eval (a procedure for T player is calculated in an analogous manner). For each block, the difference between the estimated numbers of available players' moves is calculated. The value of function $\mathrm{E_F}(B)$ (or $\mathrm{E_T}(B)$) is an upper estimate of the maximum number of moves available to the player, where B is a board or its part. It assumes that the opponent's moves will have no effect on the number of player's moves. More precisely, $\mathrm{E_F}(B) = -m\frac{m-1}{2} + \sum_{i=1}^{m} p_i$, where $p_0, p_1, \ldots p_{m-1}$ are the positions of the player's pieces on the board (block) B and $p_i \in \{0, 1, \ldots, N-1\}$. The number of the F player *non-jump* moves will be denoted by $\mathrm{G_F}(B)$ (analogously $\mathrm{G_T}(B)$ for the T player). The higher is the value of $\mathrm{G_F}(B) - \mathrm{G_T}(B)$, the better is the configuration of B for the F player. The values for the blocks created by the soft locks FF and TT are also taken into account. It is worth mentioning that in the absence of soft locks (loops in lines 4 and 6) a whole block is evaluated instead. The aim is to eliminate the relative difference between the blocks with soft locks and the blocks without them. The value of the α parameter determining the relative importance of the non-jump moves was set empirically to 0.5.

Procedure Eval(B)

1 $eval := 0$
2 **foreach** block B_i in B separated by TTFF **do**
3     $eval := eval + \mathrm{E_F}(B_i) - \mathrm{E_T}(B_i) + \alpha\,[\mathrm{G_F}(B_i) - \mathrm{G_T}(B_i)]$
4   **foreach** block B_j in B_i separated by FF **do**
5         $eval := eval + \mathrm{E_F}(B_j) - \mathrm{E_T}(B_j) + \alpha\,[\mathrm{G_F}(B_j) - \mathrm{G_T}(B_j)]$
6   **foreach** block B_j in B_i separated by TT **do**
7         $eval := eval + \mathrm{E_F}(B_j) - \mathrm{E_T}(B_j) + \alpha\,[\mathrm{G_F}(B_j) - \mathrm{G_T}(B_j)]$
8 **return** $eval$

[3] A block can be empty if the dividing lock contains the leftmost or the rightmost square of the board.

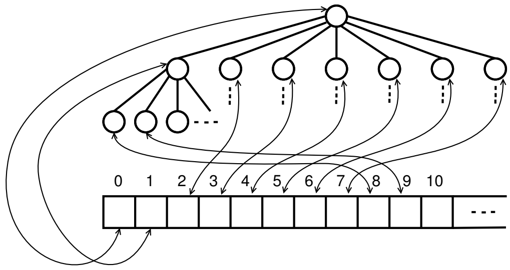


Figure 2: One-to-one correspondence between a game tree and an array.

3.2 Monte-Carlo tree search

Monte-Carlo Tree Search is useful for searching for the best move in a game. Possible moves are organized in a search tree and a number of random simulations are used to estimate the long-term potential of each move. The Monte-Carlo Tree Search method has four steps: (a) starting at the root node of the tree, select optimal child nodes until a leaf node is reached; (b) expand the leaf node and choose one of its children; (c) play a simulated game starting with that node; (d) use the results of that simulated game to update the node and its ancestors.

3.2.1 MCTS with UCT

In the presented work, plain UCT was used (Browne *et al.*, 2012). For step (a) the best child is selected by the formula:

$$\operatorname*{argmax}_{v'} \frac{Q(v')}{N(v')} + C\sqrt{\frac{2\ln N(v)}{N(v')}},$$

where v' is child of v, $N(v)$ denotes the number of simulations, $Q(v)$ denotes the number of winning simulations. An expansion strategy, step (b), is used to add one or more additional nodes to the game tree. The simulation process in step (c) replaces the need for an evaluation function. It involves playing a complete simulated game from the current node. This is done through self-play by choosing random moves for each player. As regards back propagation, step (d), it recursively updates the parents' values up to the root node. The move played by the program is the root child with the highest score (using formula from step (a)) after a fixed number of iterations.

In this paper the parameter C was set to 1. The number of expanded children was 5. The number of simulation games was set to 10. The values of these parameters were selected experimentally.

3.2.2 MCTS with fixed data storage

Our third proposed algorithm differs from the classic Monte-Carlo tree search in two fundamental ways. Instead of creating a tree online, steps (a) and (b), a fixed array[4] is used. For updating nodes, step (d), the simple minimax algorithm (Russell and Norvig, 2002) is applied, but it is executed through a parallel for-loop. Both of the alterations are aimed at dealing with severe time constraints and are not recommended when there is enough time for calculation of the next move.

Let us denote by k the maximum number of possible moves from any position, and let d denote the maximum depth (in experiments $d = 8$ was chosen). Then the size of the array (A) which maps to a game tree as shown in Figure 2, is $N = |A| = \sum_{i=0}^{d} k^i = (k^{d+1} - 1)/(k - 1)$. Thanks to such a correspondence, for the n-th array element (a virtual node) it is straightforward to find its parent index $p(n) = (n - 1)/k$ and its j-th ($0 \le j < k$) child index $c_j(n) = kn + j + 1$. The entry of the array, $A[i]$, $0 \le i < N$, is the value of a corresponding node, which represents the outcome of a position. Because the number of simulated games played for each leaf which represents an unfinished game was set to 20, $A[i] \in \{-\infty, -20, -18, -16, \ldots, 18, 20, +\infty\}$. As usual, the infinity symbol matches the end of a game: $-\infty$ a win for frogs and $+\infty$ a win for toads. Finite $A[i]$ (for

[4]We mean a static or, in other words, compile-time array. Such an array is determined in advance (when the program is compiled) and is stored in a global memory region. It is allocated before the main function runs. Monte-Carlo search trees, by contrast, are dynamic structures which need extensive use of allocating and releasing memory, which is time-consuming.

a leaf index i) counts the difference[5] between the number of toads wins and the number of frogs wins, in step (c). As in UCT, this is also done through self-play by choosing random moves for each player. For the entries which correspond to internal nodes, according to the minimax routine, the values are backing up from the leaf nodes (entries) toward the root (an element $A[0]$), one layer at a time. It is also noteworthy that the array elements which belong to the same tree level can be processed concurrently, since every value depends on a separated subarray. The operation of choosing a move is presented as Procedure FindMove. Indices on a level $0 \leq \ell \leq d$ (line 2)

Procedure FindMove(B)

```
1  for ℓ := d downto 0 do
2      for i ∈ indices_on_level(ℓ) do in parallel
3          S := a position obtained from B for i
4          if S is an achievable position then
5              if S is the toads win then
6                  A[i] := +∞
7              else if S is the frogs win then
8                  A[i] := −∞
9              else if ℓ < d then
10                 if it is the toads turn in S then
11                     A[i] := max_{0≤j<k} A[c_j(i)]
12                 else
13                     A[i] := min_{0≤j<k} A[c_j(i)]
14             else
15                 A[i] := the result of simulations
16         else
17             A[i] := a value incomparable with any number
18 if it is the toads turn in B then
19     return argmax_{1≤j≤k} A[j]
20 else
21     return argmin_{1≤j≤k} A[j]
```

can be determined directly from the set $\{(k^{\ell}-1)/(k-1), \ldots, (k^{\ell+1}-k)/(k-1)\}$. Obtaining a position S that is represented by index i (line 3) presents no difficulty, as all moves from the current position can be seen from the number of pieces[6]: the last move is made by piece number $(i-1) \bmod k$, the last but one move can be determined by the same formula substituting i for $p(i)$, etc. In line 4 the term 'achievable' is used to refer to the sequence of moves that is possible. Notice that with a certain i a position S may be unachievable due to the impossibility of moving a piece. This algorithm returns a move (in line 19 or 21) by means of the number of a piece to be moved.

In the game the number, m, of possible moves, counting those for toads as well as those for frogs, from a given position gradually decreases as the match progresses. In view of this and so as to meet strict time constraints, it is advised to increase step-by-step the value of d (we can still work on the same array A). In our program, in order to fit the search for a move within two seconds, we decided to start from $d = 5$, change it to $d = 6$ when $8 \leq m \leq 11$, $d = 7$ when $m = 7$, and $d = 8$ when $m \leq 6$.

3.3 The use of patterns and rules

The approach based on building patterns and decision rules is based on data mining (Hand, Smyth, and Mannila, 2001; Quinlan, 1979). Such a solution entails building, on the basis of a training set, a set of patterns and then a set of decision rules which is used to make decisions, i.e., to make a move, at a later stage.

As for the proposed algorithm, the process of building patterns and rules is carried out before a game starts, whereas a training set should contain a record of previously played games as well as games that other programs

[5] Notice that it has to be an even number.

[6] 0 to $k-1$ as they are seen from the left to the right.

played with each other. These could also be games played by randomized algorithms, but it is better to use records of games played by more complex programs as a basis in order to increase the quality of rules.

Patterns are constructed separately for two situations: a player will be moving frogs, or a player will be moving toads. Only parts of a game that were won by a player are selected to create patterns, i.e., for rules concerning frogs, only parts of a game that were won by the player moving frogs are analyzed, and analogously for toads. It is possible to build patterns of different lengths, but experiments were conducted for a pattern length of $|P| = n$, where n is the size of the board.

After building patterns, rules are created, i.e., for each pattern a decision is reached in the form of a move to be made. The move which is selected is the one that most often occurs in a given pattern.

When a game is played, a decision related to making every move is reached based on the same set of rules, regardless of the moment of the game. However, only rules that are best suited (i.e., those based on positions for which $f(P)$ is maximal; the definition of f is given below) to the current situation on the board are selected, and thus the move they determine can be made. A final decision is reached by first-past-the-post election among selected rules.

During the matching process only the positions of frogs F and toads T are compared, whereas empty squares □ are not taken into account. The fit is determined by using the formula:

$$f(P) = |\{i\colon 1 \leq i \leq n,\, v_i \in P,\, b_i \in B,\, v_i = b_i,\, v_i \neq \square\}|,$$

where P is a pattern and B is the current state of a board.

To create a good program an archive of games has to be used. Records of 337 games were used in the analysis. The program working, which is based on an algorithm which was connected with previously developed patterns and rules, is, most of all, characterized by a high speed of operation. When a game is played, each move is made using the same amount of time, depending on the size of the training set.

4. EXPERIMENTS

The Second Silesian University Computer Combinatorial Games Tournament took place at the Institute of Computer Science in Sosnowiec, Poland, on June 17, 2013.

Table 1: Results of the tournament

#	Author's name	R1	R2	R3	R4	R5	NBW	SOS	SOSOS
1	W. Wieczorek (MCTS)	4+	11+	5+	2+	8+	5	15	75
2	R. Skinderowicz ($\alpha\beta$)	3+	15+	6+	1-	7+	4	16	68
3	A. Mikrut	2-	9+	11+	12+	4+	4	14	73
4	P. Niegowski	1-	10+	17+	6+	3-	3	16	59
5	J. Kozak (Rules)	9+	12+	1-	7-	16+	3	14	63
6	R. Kucharski	15+	8+	2-	4-	12+	3	13	71
7	M. Gwiżdż	17+	13+	8-	5+	2-	3	13	64
8	A. Nowakowski (UCT)	0+	6-	7+	13+	1-	3	13	63
9	M. Gołab	5-	3-	15+	11+	14+	3	12	65
10	W. Kloc	11-	4-	0+	16+	13+	3	8	64
11	K. Bukowiński	10+	1-	3-	9-	15+	2	16	62
12	K. Dworak	14+	5-	16+	3-	6-	2	13	59
13	M. Cywka	16+	7-	14+	8-	10-	2	12	52
14	M. Bałchanowski	12-	17+	13-	0+	9-	2	8	56
15	P. Juszczuk	6-	2-	9-	17+	11-	1	13	66
16	K. Ciepielewski	13-	0+	12-	10-	5-	1	10	57
17	M. Środa	7-	14-	4-	15-	0+	1	9	60

As can be seen from the table, it was a Swiss-style tournament. In a single schedule, each participant played another participant (or paused) five times, so the sign + means win, and the sign - means defeat. The participants of the tournament agreed upon the following conditions: the board size, 32, the time for a single move, two seconds, and the games to be played automatically using a specially defined IP-based protocol on an AMD FX(tm)-6300 Six-Core-Processor under the Windows 7 or Fedora 17 operating system with 16 GB RAM. We decided to start from the position T□F□T□F□T□F□T□□□□□□F□T□F□T□F□T□F. The programs based on our four algorithms took first, second, fifth, and 8th places. Note, however, that due to Swiss pairings the program based on MCTS with UCT had no opportunity to play against '$\alpha\beta$' and 'Rules'.

Table 2: Results of the mini-tournament, 0.5 second per move.

Place	Program	MCTS	$\alpha\beta$	UCT	Rules
1	MCTS		68:32	97:3	99:1
2	$\alpha\beta$	32:68		81:19	94:6
3	UCT	3:97	19:81		65:35
4	Rules	1:99	6:94	35:65	

Table 3: Results of the mini-tournament, 1 second per move.

Place	Program	MCTS	UCT	$\alpha\beta$	Rules
1	MCTS		93:7	99:1	97:3
2	UCT	7:93		60:40	81:19
3	$\alpha\beta$	1:99	40:60		63:37
4	Rules	3:97	19:81	37:63	

Table 4: Results of the mini-tournament, 2 seconds per move.

Place	Program	MCTS	UCT	$\alpha\beta$	Rules
1	MCTS		77:23	87:13	90:10
2	UCT	23:77		66:34	85:15
3	$\alpha\beta$	13:87	34:66		82:18
4	Rules	10:90	15:85	18:82	

In the second series of experiments our programs played 100 games against each other. The results are depicted in Table 2, Table 3, and Table 4. From the additional experiments conducted between our two MCTS programs it transpired that 15 seconds per move is necessary before UCT begins to beat a fixed data storage approach.

5. CONCLUSIONS

In this article, we described four algorithms for playing a game quickly. The originality of two of them relies on the uncommon application of well-known methods. In the first algorithm, parallel Monte-Carlo tree search with the help of a fixed array has been proposed. In the second algorithm, the patterns and rules approach has been applied. The remaining two algorithms (described in Section 3 as the first and second), which have been given as an instance of common methods for comparison purposes, are the iterative deepening version of alpha-beta game tree search and plain UCT. On this foundation, four computer programs were prepared and submitted to a computer Toads-and-Frogs blitz tournament at the Silesian University. Our modification of MCTS undoubtedly had the best performace in the tournament and in additional experiments (all-play-all mini-tournaments).

6. ACKNOWLEDGEMENTS

The authors wish to acknowledge the anonymous reviewers for their detailed and helpful comments on the manuscript, which have contributed to describing our approaches and the results in a more precise way. This research was supported in part by Polish National Science Centre, Grant No. DEC-2011/03/B/ST6/01588, and by PL-Grid Infrastructure.

7. REFERENCES

Browne, C., Powley, E., Whitehouse, D., Lucas, S., Cowling, P. I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., and Colton, S. (2012). A Survey of Monte Carlo Tree Search Methods. *IEEE Transactions on Computational Intelligence and AI in Games*, Vol. 4, No. 1, pp. 1–43.

Erickson, J. (1996). New Toads and Frogs Results. *Games of No Chance* (ed. R. J. Nowakowski), pp. 299–310. Cambridge University Press.

Hand, D. J., Smyth, P., and Mannila, H. (2001). *Principles of data mining.* MIT Press.

Kocsis, L. and Szepesvári, C. (2006). Bandit Based Monte-Carlo Planning. *Machine Learning: ECML 2006* (eds. J. Fürnkranz, T. Scheffer, and M. Spiliopoulou), Vol. 4212 of *Lecture Notes in Computer Science*, pp. 282–293. Springer Berlin Heidelberg.

Marsland, T. A. (1986). A review of game-tree pruning. *ICCA Journal*, Vol. 9, No. 1, pp. 3–19.

Quinlan, J. R. (1979). Discovering Rules by Induction from Large Collections of Examples. *Expert Systems in the Micro-Electronic Age* (ed. D. Michie), pp. 168–201. Edinburgh University Press.

Russell, S. and Norvig, P. (2002). *Artificial Intelligence: a modern approach.* Prentice Hall.

Thanatipanonda, T. A. (2008). Further Hopping with Toads and Frogs. http://arxiv.org/pdf/0804.0640v1.pdf.

CHESS ENDGAME NEWS

G.M^c^C. Haworth[1]

Reading, UK

This note includes some endgame reflections on the last World Chess Championship, an update on the search for the longest decisive games between computers, and a brief mention of the sets of endgame table (EGT) statistics recently received from Yakov Konoval (2013) and from Victor Zakharov (2013) for the Lomonosov team.

Following hard on the heels of Nunn's (2013) review of instructive errors in the analysis of KRPPKRP, Carlsen-Anand, FCWM 2013 game 5, arrived at this very ending with position 53w, 8/8/8/2p1k3/P6R/1K6/6rP/8 w. The Lomonosov EGTs say 'mate in 33m' starting **53. a5′′′′** (only winning move) **Kd6′** (equi-DTM-optimal) **54. Rh7′′** (unique optimal) **Kd5′′ 55. a6′′′′ c4′′**, all duly played. After **56. Kc3 Ra2′′ 57. a7′′**, Black hastened the end with **57. … Kc5** (-8m; Ra4′′) and resigned after **58. h4′′**. A likely continuation was the DTM-optimal 59. h5′ Ra3′ 60. Kc2′ c3′ 61. h6′ Ra2′′ 62. Kxc3′ with a new Queen coming onboard around move 70.

Anand later defined game 5 as a turning point in the match, and immediately went two down after game 6 where he strayed in a 10-man R-endgame with **60. Ra4?** and resigned with nine men on the board. Game 10 ended in KKN with bare Kings alongside an echo of Carlsen's Knight sacrifice. Needing only a draw, he had untypically ignored a somewhat more promising line at 8/1p2k3/p3pN1p/P1K2pp1/2P2P2/1P2n1PP/8/8 w: 46. Nh5 Kf7 47. Kb6 Kg6 48. Kxb7 Kxh5 49. c5 gxf4 50. gxf4 e5 51. c6 exf4 52. c7 f3 53. c8=Q f2 54. Qe8+ Kh4 55. Qxe3 f1=Q 56. Qxh6+ Kg3° 57. Qxa6 Qxa6+ 58. Kxa6 **{KPPPKP, =}** f4 59. Kb6 f3 60. a6 f2 61. a7 f1=Q 62. a8=Q **{KQPPKQ**, =} Qf2+ 63. Kb5 Qe2+ 64. Kb4 Kxh3 **{KQPKQ**, =} 65. Qc8+ =.

Carlsen points out that computers and computer databases have made opening theory more widely available, levelling the initial playing field and leading to only marginal advantage in the middlegame by the first time-control. If so, we may look forward to many more games where subtle advantages are accumulated slowly and result in a display of fine endgame technique and a hard-earned victory.

Hernandez (2013) notes some decisive computer games which nudge up the length-records (Haworth, 2013a/c) and/or break the record for games extended by DTM-minimaxing play inferred from available EGTs:

a) STRELKA -v- SCORPIO (2013-04-09, E15): ending at KQPPKQN position 301w, theoretically drawn: the indicated '0-1' result may be an error, another reason to ignore a long game as a record-holder,

b) NAUM_4.2 -v- TORNADO_4.25 (2011-01-08, A84): ending at KPPKPP position 300w (*dtm* = 15m): thus, the extrapolated length (to mate) from p300w is 314m/627p,

c) HOUDINI_3_PRO -v- KOMODO_6 (2013-06-11, D23): ending at KQRKQP p296b (*dtm* = -36m): the extrapolated length (to mate) from p296b is 332m/663p.

It seems clear that there has been and perhaps still is a ceiling imposed by technology, including that of Chessbase, which makes it difficult if not impossible to record games of more than 300 moves. A pity, as they have surely been played between computers and may be classic battles with interesting endgames.

Yakov Konoval (2014) has filed three sets of Depth to Conversion (DTC) statistics with the author:

- a complete set of statistics for the 645 'White win' EGTs of 6-man chess:[2]
  n.b., the maxDTC 6m decisive position is not in KRNKNN but a KRRPKQ loss in 486 plies,
- statistics for 'White win' 7m chess covering all 680 P-less and 460 of the 1,070 P-ful EGTs[3] and
- statistics for a further 285 7m sub-endgames with specific square-colour profiles for the Bishops.

The EGTs themselves are not available via a query-service on the web and his work with Marc Bourzutschky, dating from 2004, deserves to be more available and better known. This journal has frequently reviewed the results they have highlighted in some six articles in *EG*, the endgame studies magazine.

---

[1] The University of Reading, Berkshire, UK, RG6 6AH. email: guy.haworth@bnc.oxon.org.

[2] 295 P-less and 350 P-ful EGTs, with the usual caveat - positions with non-null castling rights not included.

[3] There are 140 5-2, 200 4-3, 210 5-2p and 325 4-3p endgames. The caveat here - P-promotion was restricted to P=Q.

Chess Endgame Records (CER) is an evolving, annotated dataset (Haworth, 2013b) and summarises, for example as in Figure 1, the known and candidate maxDT*x* records to date. It now includes the 6-man maxDTC records established by Konoval and the 7m maxDTM records established by the Lomonosov team (Zakharov, 2013). The Lomonosov DTM data reinforces the author's belief (Haworth, 2013d) that there is a discernable trend in the growth of maxDTM as the number of men on the board grows. There are also confirmed and candidate DTC/Z records for some parts of 7-man chess.

The dataset is also a partial reconstruction of the history of EGT generation since the concept was first formulated (Bellman, 1964; Knuth, 1968). It notes record achievements of the past, many of which exploited the available technology of the time to the limit. Given that thirty years have produced computers with a million times more power and memory, it is easy to forget this.

m	w-b	DTC: depth in plies			DTM: depth in plies			DTZ: depth in plies		
		P-less	all	P-ful	P-less	all	P-ful	P-less	all	P-ful
3	2-1	KRK -32	KPK -38	KPK -38	KRK -32	KPK -56	KPK -56	KRK -32	KRK -32	KPK -20
4	2-2	KQKR -62	KQKR -62	KQKP -53	KRKN -80	KPKR 85	KPKR 85	KQKR -62	KQKR -62	KQKP -53
	3-1	KBNK -66	KBNK -66	KNPK -44	KBNK -66	KBNK -66	KPPK -64	KBNK -66	KBNK -66	KNPK -26
	all	KBNK -66	KBNK -66	KQKP -53	KRKN -80	KPKR 85	KPKR 85	KBNK -66	KBNK -66	KQKP -53
5	2-3	KQKBB -142	KQKRP 157	KQKRP 157	KQKBB -162	KQKRP 207	KQKRP 207	KQKBB -142	KQKRP 151	KQKRP 151
	3-2	KBNKN 153	KNNKP 228	KNNKP 228	KBNKN 213	KPPKP -254	KPPKP -254	KBNKN 153	KNNKP 164	KNNKP 164
	4-1	KNNNK -42	KNNNK -42	KBBPK -32	KBNNK -68	KBNNK -68	KPPPK -66	KNNNK -42	KNNNK -42	KBBPK -24
	all	KBNKN 153	KNNKP 228	KNNKP 228	KBNKN 213	KPPKP -254	KPPKP -254	KBNKN 153	KNNKP 164	KNNKP 164
6	2-4	KQKBBN 125	KQKBNP -384	KQKBNP -384	KQKBBN -228	KPKBNP 447	KPKBNP 447	KQKBBN 125		
	3-3	KRNKNN 485	KRNKNN 485	KQPKRB -272	KRNKNN 523	KRNKNN 523	KRPKNN 505	KRNKNN 485	KRNKNN 485	
	4-2	KRRNKQ -202	KRRPKQ -486	KRRPKQ -486	KRBNKQ 241	KRRPKQ -506	KRRPKQ -506	KRRNKQ -202	KRRPKQ 383	KRRPKQ 383
	5-1	KBBBNK -27	KBBBPK -31	KBBBPK -31				KBBBNK -27		
	all	KRNKNN 485	KRRPKQ -486	KRRPKQ -486	KRNKNN 523	KRNKNN 523	KRRPKQ -506	KRNKNN 485	KRNKNN 485	KRRPKQ 383
7	2-5	KQKBBBB 131	? KQKBNPP -202	? KQKBNPP -202	KQKRBBN 239	KQKBBNP -486	KQKBBNP -486	KQKBBBB 131		
	3-4	KQNKRBN -1,034	? KQNKRBN -1,034	? KRBKBNP -412	KQNKRBN -1,090	KQPKRBN 1,097	KQPKRBN 1,097	KQNKRBN -1,034	? KQNKRBN -1,034	
	4-3	KQBNKQB -660	? KQBNKQB -660	? KRNPKRB 529	KQBNKQB -690	KBNPKBP 691	KBNPKBP 691	KQBNKQB -660	? KQBNKQB -660	?KRNPKRB 519
	5-2	KBNNNKQ -448	? KNNNPKQ 461	? KNNNPKQ 461	KBNNNKQ -464	KRBBPKQ 799	KRBBPKQ 799	KBNNNKQ -448		
	all	KQNKRBN -1,034	? KQNKRBN -1,034	? KRNPKRB 529	KQNKRBN -1,090	KQPKRBN 1,097	KQPKRBN 1,097	KQNKRBN -1,034	? KQNKRBN -1,034	?KRNPKRB 519

Figure 1. maxDT*x* wins for White (Haworth, 2013b), '-' indicating 'loser to move'.

References

Bellman, R.E. (1964). Dynamic Programming and Markovian Decision Processes, with particular application to Baseball and Chess. In Applied Combinatorial Mathematics (ed. E.F. Beckenbach), John Wiley, pp. 221-236.

Haworth, G. McC. (2013a). Chess Endgame News. *ICGA Journal*, Vol. 36, No. 3, pp. 143-145.

Haworth, G. McC. (2013b). Chess Endgame Records. Dataset at http://centaur.reading.ac.uk/34268/.

Haworth, G. McC. (2013c). Chess Endgame News. This paper plus datafiles. http://centaur.reading.ac.uk/36275/.

Haworth, G. McC. (2013d). Haworth's Law. *ICGA Journal*, Vol. 36, No. 4, p. 230.

Hernandez, N. (2013). Private communication of long computer-computer games.

Knuth, D.E. (1968). *The Art of Computer Programming*, Vol. 1, esp. pp. 272-273 in the 3rd edition.

Konoval, Y. (2013). Private communication of 6- and 7-man DTC EGT statistics.

Nunn, J. (2013). Discoveries in R+2P vs R+P endings. *ICGA Journal*, Vol. 36, No. 3, pp. 139-142.

Zakharov, V. (2013). Private communication of 'MVL' Lomonosov 7-man DTM EGT statistics.

HAWORTH'S LAW

G.M^cC. Haworth[1]

Reading, UK

The latest 'Depth to Mate' results from the Lomonosov team (Zakharov, 2013) find a maxDTM 7-man wtm win in KQPKRBN of 1,097 plies, i.e. of 549 winner's moves. They therefore add one data point to an already suggestive trend of log(maxDTM) against k, the number of men on the board. Figure 1 is a plot of the data (Haworth, 2013) showing the actuals for a 3- to 7-man chess, the best least-squares linear fit[2] to these points, and the extrapolation of that 'fit' to 10-man chess with 2σ, 97% probability, confidence levels.

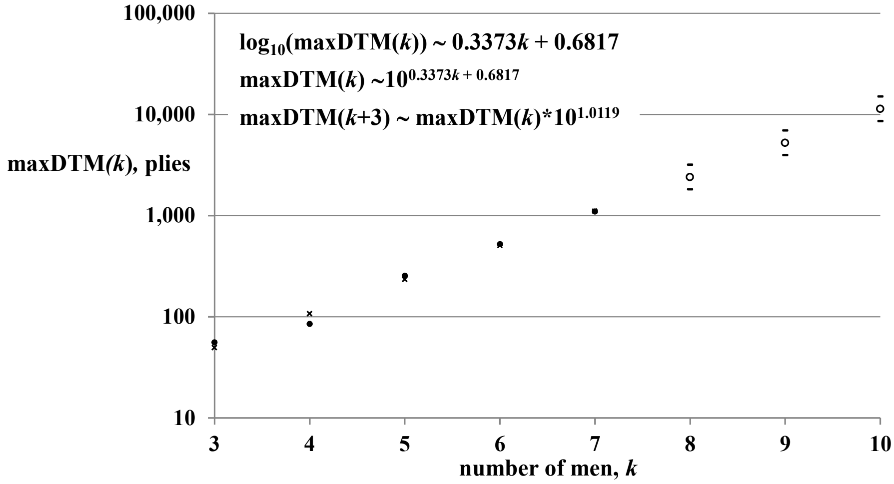


Figure 1. The maxDTM(k) trend: actuals, best linear fit, predictions and 2σ confidence intervals.

Here are some of the conjectures which may be made, using the following notation:

$E \equiv WB$, an endgame with White force W and Black force B,
$Em \equiv WmBm$, endgame E with man m added to both sides,
maxDTM(E) ≡ the maximum DTM in plies of the White wins in E ('0' if there are no wins), and
maxDTM(k) ≡ max{maxDTM(E) | E is a k-man endgame}

1) if $k \geq 3$, maxDTM(k+1) > maxDTM(k),
2) if $k \geq 3$, a maxDTM k-man position p_k may be modified to a position p_{k+1} with greater DTM depth: the side which does not have the move may often be imagined to have just captured a man,
3) if $k \geq 3$, there is a k-man endgame E and man m such that maxDTM(Em) ≥ maxDTM(E),
4) the linear trend above will continue for some time, i.e., 'Three more men: maxDTM times ten!"

With Moore's Law in mind, the last conjecture was dubbed *Haworth's Law*, as it were, *en passant* by a visiting Thomine Stolberg-Rohr WFM. It is certainly a prediction like Moore's Law rather than a provable, physical law. However, it is not a self-fulfilling prophecy as many argue Moore's Law is. The rules of the game have determined those deep wins and losses already. For 8/9/10-man chess, the model gives a 50% probability of decisive results in 2400^+/ 5220^+/11340^+ plies and 2σ-predictions of results in 1810^+/3940^+/8570^+ plies. It gives a 90% probability of an 8m result in 2000^+ plies and an 80% probability of a 10m result in 10000^+ plies. The model at least challenges us to consider why this might be and how long the trend will continue.

References

Haworth, G. M^cC. (2013). Chess Endgame Records. Dataset at http://centaur.reading.ac.uk/34268/.

Zakharov, V. (2013). Private communication of 'MVL' Lomonosov 7-man DTM EGT statistics.

[1] The University of Reading, Berkshire, UK, RG6 6AH. email: guy.haworth@bnc.oxon.org.
[2] The best-fit quadratic polynomial reduces the 'linear' residual error by only 6% and gives even higher predictions for the 8//9/10-man maxDTM. The best cubic and quartic fits clearly give overfitted models which are not credible.

REVIEW

FROM TO αβ TO ABCD AND SMAB

Solving Games and All That
Abdallah Saffidine
PhD Thesis, Université Paris-Dauphine 2013, 175 pp.[1]
Reviewed by Dap Hartmann

As the foundation for this thesis, Abdallah Saffidine develops a framework for deterministic two-player games with perfect information and two outcomes, to represent best-first search (BFS) algorithms such as Proof Number search (PNS), Monte-Carlo Tree Search (MCTS) and the Product Propagation (PP) algorithm. "PNS is a best first search algorithm that enables to dynamically focus the search on the parts of the search tree that seem to be easier to solve". PNS has been applied successfully in many games, especially in 'difficult' games such as Checkers, Shogi and Go. The problem with PNS is that it is resource-intensive because the entire game tree needs to be kept in memory. "The basic idea in MCTS is to evaluate whether a state *s* is favourable to *Max* via Monte Carlo playouts in the tree below *s*. A Monte Carlo playout is a random path of the tree below s ending in a terminal state." MCTS has proven to be very successful in games such as Go where progress had been slow because of the large branching factor and the extensive horizon effects. Product Propagation was a relatively new concept to me. "PP is a way to backup probabilistic information in a two-player game tree search. It has been advocated as an alternative to minimaxing that does not exhibit the minimax pathology." Minimax pathology, which was discovered independently by Dana Nau and Don Beal some 35 years ago, is the counter-intuitive effect that deeper minimax searches do not always result in better play. "PP was recently proposed as an algorithm to solve games, combining ideas from PNS and probabilistic reasoning." Although the PP algorithm performs well in Go does not do so well in other games, such as Shogi. However, Saffidine shows examples of three games in which PP outperforms the more traditional search algorithms: The game of Y (a connection game invented by Claude Shannon), Domineering, and Nogo (a misère version of Go in which the first player to capture loses).

In Chapter 3, Saffidine adapts his framework to two-player games with multiple outcomes which results in a Best First Search (BFS) framework. Using a principled approach, he creates a 'multi-outcome information scheme' which he calls 'multization', not to be confused with the Multization app which stands for 'Multiplication X Memorization', a fancy version of multiplication tables. Saffidine uses multization to generalize PNS and PP for multi-outcome games. The resulting Multiple-Outcome Proof Number Search (MOPNS) algorithm is applied to two games: Connect Four and Woodpush. Although Connect Four was already solved in 1988 by Victor Allis and James Allen, it still provides an interesting benchmark to test search algorithms. For 89% of the 256 4x5 positions that were tested, MOPNS needed fewer nodes than PNS but at the expense of requiring 16% more time. The same pattern was found for the 625 5x5 positions that were tested and in which MOPNS needed fewer nodes in 65% of the cases using 14% more time. For Woodpush, a recent game that involves forbidden repetition of the same position, the results were comparable to the performance exhibited in Connect Four: fewer nodes at the price of more time.

Chapter 4 investigates the relationship between Multi-agent Modal Logic K (MMLK) and sequential game search and suggests several new model checking algorithms. Saffidine shows how the MMLK Model Checking framework can be used to develop new research in game tree search. He focusses on turn-based games with perfect and complete information. Not just two-player games, such as Chess and Go, but also single-player games (puzzles) like Rubik's Cube and Sokoban, and multiplayer games such as Chinese Checkers (a variation of the game Halma that is played by two, three, four or six people). Saffidine suggests several new model checking algorithms for MMLK and proves that one of them (Minimal Proof Search - MPS) is correct and that, under certain conditions, it minimizes a generic cost function. However, MPS has some limitations because it is a memory-intensive best-first search algorithm that currently cannot make use of transpositions.

The final chapter deals with two-player zero-sum games with simultaneous moves, the so-called stacked-matrix games. For this domain, Saffidine developed the Simultaneous Move Alpha-Beta (SMAB) algorithm which is a generalized version of the alpha-beta pruning algorithm and is described as "a depth-first search algorithm

[1] This thesis can be downloaded from:
http://cgi.cse.unsw.edu.au/~abdallahs/Papers/2013/Solving%20Games%20and%20All%20That.pdf

[which loops] through all joint action pairs first checking trivial exit conditions and if these fail, proceeding with computing optimistic and pessimistic bounds for the entry in questions, and then recursively computing the entry value." SMAB involves a large computational overhead because it needs to solve Linear Programs. Saffidine developed heuristical optimizations to speed up this process. The result was an algorithm that solved Goofspiel faster than alternative methods (backward induction and sequence form solver). Goofspiel (also known as The Game of Pure Strategy or GOPS) is a card game in which three of the four suits are used and the players move simultaneously. Saffidine experimented with various numbers of cards per suit to analyze the pruning efficiency as a function of game-tree size.

Saffidine also discusses the Alpha-Beta (Considering Durations) algorithm (ABCD), an efficient heuristic algorithm for Real-Time Strategy games which involve simultaneous moves under tight time constraints. One such game is Starcraft which, according to Wikipedia, "[m]any of the industry's journalists have praised […] as one of the best and most important video games of all time, and for having raised the bar for developing real-time strategy games". Unfortunately, I am totally ignorant of this type of game, so I will merely cite Saffidine on his future goal in this domain: "Our next steps will be to integrate ABCD search into a STARCRAFT AI competition entry to gauge its performance against previous year's participants, to refine our combat model if needed, to add opponent modelling and best-response-ABCD to counter inferred opponent combat policies, and then to tackle more complex combat scenarios."

'Solving Games and All That' is an excellent thesis with a solid game-theoretical framework that uses 42 definitions, 8 theorems, 36 propositions, 1 lemma, 11 algorithms, 15 examples and 7 remarks, distributed over 5 chapters in the space of 175 pages. It is well structured and written in a smooth style that reads like a charm. The definitions and descriptions of both well-known concepts and new notions that Saffidine provides are exemplary. So much so that I have liberally quoted from his text because I cannot phrase many of Saffidine's descriptions better myself.

Overview of Playing Hall in Yokohama.

INFORMATION FOR CONTRIBUTORS

Submission of material

Contributions to the Journal of up to 4000 words are welcome in any form. Short contributions (2000 - 3000 words) are preferred. Longer papers of striking originality and cogency may be considered, provided the authors consent to publication in instalments. Authors are urged to supply full references, including the names and location of publishers and ISBN or ISSN numbers where applicable.

While any form is welcome, efficiency of production imposes an editorial preference for some forms usual in electronic communications. The preferred forms are (in that order) Word (version 7.0 or convertible), LaTeX (information on the ICGA LaTeX style file can be found at: http://icga.uvt.nl/?page_id=277).

Figures not conveniently accommodated in any of the above formats may be submitted either in PostScript format or as camera-ready hard copy. We urge contributors not to hesitate to submit their material in the form which most clearly represents the authors' wishes as to lay-out and presentation. More information is available at our homepage at http://www.icga.org.

Abstracting and indexing

Contributors may be interested to know that the *ICGA Journal* (and previously the *ICCA Journal* as of Vol. 10, No. 1 (1987)), is a source for Thomson Scientific for inclusion in the CompuMath Citation Index®, the Science Citation Index Expanded®, ISI Web of Science®, SciSearch®, and Personal Alert®. Paper abstracts and author information are accessible online through ISI Web of Science®.

The journal is also a source of information for R.R. Bowker for inclusion in the International Serials Database which is a source for Ulrich's Periodicals Directory™. Since 2006, the *ICGA Journal* is included in the abstract and citation database Scopus® by Elsevier B.V.

The Editorial Board

Broadening the scope of the Journal from chess to games has resulted in an extended Editorial Board, which now possesses specific knowledge on a variety of games. We mention Amazons, Backgammon, Bridge, Checkers, Chess, Chinese Chess, Chinese Dark Chess, Clobber, Dots and Boxes, Draughts, Ein Stein Würfelt Nicht, Go 9x9, Go 13x13, Go 19x19, Havannah, Hex, LOA, NoGo, Nonograms, Othello, Phantom Go, RoShamBo, Shogi, Sudoku and Surakarta. Contributors are encouraged to bring their submissions to the attention of the Editor-in-Chief via a game-knowledgeable member of the Editorial Board. Their names and email addresses are:

NEWS, INFORMATION, TOURNAMENTS, AND REPORTS

THE 8TH COMPUTERS AND GAMES CONFERENCE

H.J. van den Herik[1]*, H. Iida*[2]*, and A. Plaat*[1]

Yokohama, Japan

The 8th Computers and Games conference (CG2013) conference was held in Yokohama, Japan over a three-day period from August 13 to August 15, 2013. The conference was held in conjunction with the 20th World Computer-Chess Championship and the 17th Computer Olympiad. The venue for all three events was the Hiyoshi Campus of Keio University in Yokohama. All presentations were also followed by livestream video. The conference report is provided by three authors, viz. Ingo Althöfer (day 1), Simon Viennot (day 2), and Richard Lorentz (day 3). Following Althöfer's contribution published in the September 2013 issue of the *ICGA Journal* (pp. 170-171), we complete our report of the conference below.

Conference Report CG2013, day 2

Simon Viennot[2]
Kanazawa, Japan

Havannah and TwixT are PSPACE-complete by **Edouard Bonnet, Florian Jamain** and **Abdallah Saffidine**. The first paper was presented by Abdallah Saffidine. The authors showed that the games Havannah and TwixT are in the PSPACE-complete class of complexity. This is not unexpected since both Havannah and TwixT are connection games, somewhat similar to Hex, known to be PSPACE-complete since 1981. The proof relies on a reduction, by encoding some games already known to be PSPACE-complete in the target games. Interestingly, the proof of this theoretical problem involves mainly beautiful figures showing how Hex can be encoded in the game of TwixT, and how Generalized Geography can be encoded into Havannah.

Anomalies of Pure Monte Carlo Search in Monte Carlo Perfect Games by **Ingo Althöfer** and **Wesley Michael Turner**. This paper presented by Ingo Althöfer showed how anomalies can happen in pure Monte-Carlo search even in the case of Monte-Carlo perfect games. Monte-Carlo perfect games are games where a pure Monte-Carlo search (also called sometimes *flat Monte-Carlo search*) converges to the perfect move for any position of the game if the search time is unlimited. The surprising result is that if the search time is limited, then you have no guarantee that pure Monte-Carlo will give better results with more search time, even on this class of Monte-Carlo perfect games. In fact, concrete Monte-Carlo perfect games can be constructed to make a pure Monte-Carlo search with less simulations arbitrarily stronger than a search with more simulations.

Solution Techniques for Quantified Linear Programs and the links to Gaming by **Ulf Lorenz, Thomas Opfer** and **Jan Wolf**. Completing Session 4, Thorsten Ederer presented this talk on Quantified Linear Programs (QLPs). QLPs are an extension of Linear Programs, where some variables are universally quantified ("for all'" quantifiers) instead of only existentially quantified ("there exists'" quantifiers). QLPs can be interpreted as games between two players, and the authors show that it is possible to combine algorithms from the QLPs field (Nested Benders Decomposition) and algorithms from the computer game field (α-β search). The resulting α-β-NBD algorithm finds solutions to QLP instances more quickly than the standard NBD algorithm.

Cylinder-Infinite-Connect-Four except for Widths 2, 6, and 11 is Solved: Drawn by **Yoshiaki Yamaguchi, Tetsuro Tanaka**, and **Kazunori Yamaguchi**. Session 5 started with a talk on the classical game of Connect-Four, played on an infinite cylinder. Yoshiaki Yamaguchi presented a proof that cylinder-infinite Connect-Four is solved for almost all widths (except 2, 6 and 11), and the result is *Draw* under perfect play. The proof relies on specific tilings of the infinite cylinder showing that both players have strategies to avoid losing. The talk spurred discussions about the case of width 2.

Automatic Generation of Chinese Dark Chess Opening Books by **Bo-Nian Chen** and **Tsan-sheng Hsu**. The talk was presented by Bo-Nian Chen. Dark Chess, also known under its Chinese name "Banqi'", is a variant of

[1] Tilburg center for Cognition and Communication (TiCC), Tilburg University, Tilburg, The Netherlands. Email: {H.J.vdnHerik, A.Plaat}@tilburguniversity.edu
[2] Japan Advanced Institute of Science and Technology (JAIST), Kanazawa, Japan. Email: sviennot@jaist.ac.jp

chinese chess, where the pieces are turned face-down at the beginning of the game. The game gradually turns from incomplete-information in the opening to complete-information in the end. In a work with Tsan-sheng Hsu, Bo-Nian Chen shows how to evaluate automatically some opening strategies like attacking the opponent's pieces or increasing the mobility of the player's pieces. This evaluation leads to the first known opening book for the game of Dark Chess.

Improving Best-Reply Search by **Markus Esser, Michael Gras, Mark H.M. Winands, Maarten P.D. Schadd**, and **Marc Lanctot**. In Session 6, Marc Lanctot introduced a new algorithm BRS^+ for multi-player games. Best-Reply Search (BRS) is a recent algorithm that can be used to accelerate the search in multi-player games by flattening the opponent's actions to only one strongest opponent at each turn. However, the BRS algorithm can lead to invalid turn sequences, and also gives an unfair advantage to the root player. The authors showed that this weakness of BRS can be corrected by using heuristics to order the opponent moves and generate a default sequence of moves instead of a sequence of passes for the opponents flattened by the search. The resulting BRS^+ algorithm is significantly stronger than all other algorithms in the game of Four-Player Chess.

Scalable Parallel DFPN Search by **Jakub Pawlewicz** and **Ryan B. Hayward**. The last talk of the day was given by Jakub Pawlewicz about the problem of parallelizing Depth-First Proof Number Search (DFPN). The authors described a new parallel algorithm called Scalable Parallel Depth-First Proof Number Search (SPDFPN). This is a parallelized version of DFPN, inspired by the techniques of virtual win/loss used to parallelize Monte-Carlo Go. A virtual win/loss state is assigned to the node currently searched by a thread, and accordingly virtual proof and disproof numbers are added on the path from the root to this node. In effect, this forces each thread to search different nodes, in an order close to the one of a single-thread DFPN. In experiments on the game of Hex, SPDFPN was shown to scale well even with 16 threads.

Conference Report Day 3

Richard J. Lorentz[1]
Northridge, California, USA

Efficiency of Static Knowledge Bias in Monte-Carlo Tree Search by **Simon Viennot** and **Kokolo Ikeda**. The conference's Session 7 began with a presentation by Simon Viennot. Biasing the random playouts (simulations) in MCTS is a well-known, fairly well understood, and reasonably standard technique in MCTS-based programs. However, biasing the tree traversal is less well formalized and is little mentioned in the literature even though techniques such as progressive widening and knowledge bias are commonly used. The authors' stated goal in this article is "to give a more detailed description of these two ideas and show experimental evidence that they can be implemented and tuned efficiently."

Developments on Product Propagation by **Abdallah Saffidine** and **Tristan Cazenave**. The talk was given by Tristan Cazenave. Product Propagation is a technique for backing up probabilistic information in a game tree and the authors propose using this technique for solving games. They test their ideas against other programs where, for the sake of uniformity, none of the programs use any domain specific knowledge. Of course they realize that using such knowledge would improve any of the programs, but they state: "Nevertheless, we believe that showing that a generic and non-optimized implementation of PP performs better than generic and non-optimized implementations of PNS, MCTS, or αβ in a variety of domains provides good reason to think that the ideas underlying PP are of importance in game solving."

Analyzing Simulations in Monte-Carlo Tree Search for the Game of Go by **Sumudo Fernando** and **Martin Müller**. Since neither author could attend the conference the contribution was presented by Ryan Hayward and Jakub Pawlewicz. The main idea of the paper is to provide information about how playouts work in MCTS and this was done in the context of the Go-playing program FUEGO. Various combinations of playout policies were tried where results were measured in terms of the number of blunders made. After a short break Session 8 began.

1 Department of Computer Science, California State University, Northridge, CA 91330-8281, USA. Email: lorentz@csun.edu

Material Symmetry to Partition Endgame Tables by **Abdallah Saffidine, Nicolas Jouandeau, Cédric Buron**, and **Tristan Cazenave**. The talk was presented by Abdallah Saffidine. Using endgame databases is a well known and important technique found in many game playing programs. One of the main problems with them is that they can require enormous amounts of memory. One way to reduce these memory requirements is to recognize symmetries so that only one of the many symmetrical positions need be stored in the database, but recognizing symmetries is not always easy. According to the authors: "In this paper, we propose a principled framework to detect material symmetry and show how it can be applied to three games: chinese dark chess, dominoes, and skat. At the core of our method lies the sub-graph isomorphism problem which has been extensively studied in computer science."

Further Investigations of 3-Member Simple Majority Voting for Chess by **Kristian Spoerer, Toshihisa Okaneya, Kokolo Ikeda**, and **Hiroyuki Iida**. The presentation was given by Kristian Spoerer. The authors studied the advantage of having three chess programs (of similar strength) voting on the best move, i.e., "Will such a system outperform the best of the three programs when acting alone?"(The short answer is "yes".) They also try to understand under what conditions this will be true. Again, the short answer is that there are two conditions. Condition 1: "group members should be almost equal in strength whilst still showing significant strength difference", and Condition 2: "denial percentage of the leader's candidate should depend on the strength of the members." Session 9, the last session of the conference, began after lunch.

Optimal, Approximately Optimal, and Fair Play of the Fowl Play Card Game by **Todd Neller, Marcin Malec, Clifton Presser**, and **Forrest Jacobs**. The talk was given by Todd Neller. Fowl Play is an interesting card game for two or more players with a mix of strategy and luck. The 48-card deck contains 42 "chickens" and 6 "wolves". One takes cards hoping to see chickens. A player's turn ends either when he chooses to stop taking cards and collects a point for each chicken he has taken or when he picks a wolf, in which case he gets no points et all. Playing well seems to require a delicate balance of both greed and caution. Among other results, the authors prove an optimal playing strategy, noting that it is quite different from a score-maximizing strategy.

The final paper of the conference was one of two papers to share the Best Paper Award (see Vol. 36, No. 2, p. 169). *Dependency-Based Search for Connect6* by **I-Chen Wu, Hao-Hua Kang, Hung-Hsuan Lin, Ping-Hung Lin, Ting-Han Wei, Chieh-Min Chang**, and **Ting-Fu Liao**. The paper was presented by Ting-Han Wei. Connect6 is a game similar to Go-Moku and Renju only players place two stones per turn rather than one and they must get 6 in a row rather than 5. (The first player only plays one stone.) They show how dependency-based search can be used with threat-based search to improve NCTU6, their already very strong Connect6 playing program. They propose 4 dependency-based search strategies and show that one of them "...yields a speedup factor of 4.12 on average, and up to 50 for certain hard positions". Some nice results in a well written paper provided us with an excellent choice for the Best Paper Award. I would be remiss if I did not point out that this was yet another interesting, stimulating, and well run event by everyone involved in the ICGA.

COMPUTER GAMES AND INTELLIGENCE WORKSHOP

K. Spoerer[1]
Kanazawa, Japan

The *Computer Games and Intelligence Workshop* was held as part of the *ICGA events in Yokohama, 2013*, in Japan on 15 August 2013. The workshop was held in conjunction with the 8th International Conference on Computers and Games (CG2013), and was organized by ICGA, and co-organized by IPSJ-SIG-GI. Three times a word of thanks (1) to the organizing committee (Takeshi Ito, Kristian Spoerer, Hitoshi Matsubara and Hiroyuki Iida), (2) to the contributors for sharing their new research at the workshop, and (3) to the session chairs Junji Nishino and Takenobu Takizawa.

The workshop aimed for a stimulating discussion of *new emerging research* in the field of Games and Intelligence. In total there were nine contributions, with talks from 6 Japanese researchers, one German, one Chinese, and one Taiwanese. Thanks to strong efforts, all talks and discussions were conducted in English. The workshop was split into two sessions, with five talks in the first session, and the remaining four talks in the second session.

[1] JAIST, Kanazawa, Japan. Email: kristian@jaist.ac.jp

In the first session Hiroyuki Iida gave a talk about adding synchronism into combinatorial games. Next, Ingo Althöfer presented some findings on random LEGOTM structures. Then, Xiong Shuo discussed his ideas on 鎖国 (sakoku) "Seclusion From the Outside World" in China and Japan. Lung-Ping Chen talked about processor allocation in parallel game tree search. Finally, Junji Nishino finished the first session by describing a measurement, for imperfect information games, for how dependent profit is on the distribution of actual situations at that point in time.

In the second session Takenobu Takizawa talked about contemporary computer Shogi. Afterwards Taichi Ishitobi presented his work on automatic composition of checkmate problems in Shogi. Then Takafumi Nakamichi gave a talk on (1) studying an AI for Shogi by using human records, and (2) analysing human expertise in Shogi. Finally, Yuichiro Sato provided a mathematical treatment of the consultation algorithm.

THE BRAIN AND MIND-SPORTS COMPUTER OLYMPIAD

Yokohama, Japan
August 12 – 18, 2013

Jaap van den Herik[1], Hiroyuki Iida[2], Aske Plaat[1], and Johanna Hellemons[1]

Kanazawa, Japan Tilburg, The Netherlands

The Keio University was the location of the Brain and Mind-Sports Computer Olympiad, held in Yokohama, Japan. The Olympiad was made possible by the generous sponsorship of the Brain and Mind-Sports Foundation, JAIST, TiCC, ICGA, and Digital Games Technology. The organisation of the Olympiad was in the hands of Johanna Hellemons (chair), Hiroyuki Iida, Setsuko Asakura, H.Jaap van den Herik, and David Levy.

As usual, the Computer Olympiad consisted of many different competitions, each having their own programs and their own rules. Chess is organised as a separate event. The WCCC 2013 is won by JUNIOR and the WCSC 2013 is won by HIARCS, the World Speed-Chess Championship by SHREDDER (see Vol. 36, No. 2, pp. 151-158, pp. 159-165, and p. 158 respectively. The 17th Computer Olympiad hosted the following twenty-one games. In Table 1 we list the games, winning program, and their authors.

Game	Winning Program	Author(s)
Amazons	INVADER	Richard Lorentz
Chinese Chess	SHIGA	Shi-Jim Yen
Chinese Dark Chess	DARKKNIGHT	Weng-Jie Tseng
Chu Shogi	HACHU	Harm Geert Muller
Clobber	PAN	Johan de Koning
Connect6	MOBILE6	Ting-han Wei
Dots and Boxes	BITDB	Fu JiGao and Wang Bo
10×10 Draughts	TDKING	Ton Tillemans
Ein Stein Würfelt Nicht	PROPHET_WT	Lin Tang
Go 9×9	ZEN	Hideki Kato and Yoji Ojima
Go 13×13	ZEN	Hideki Kato and Yoji Ojima
Go 19×19	ZEN	Hideki Kato and Yoji Ojima
Hex	MOHEX	Broderick Arneson, Ryan Hayward, and Philip Henderson
LOA	MC-LOA	Mark Winands
Mahjong	LONGCAT	Ting-Han Wei
NoGo	HAPPYNOGO	Ting-Han Wei
Nonogram	LALAFROGKK	Chen-Kang Kao
Phantom Go	GOLOIS	Tristan Cazenave
Shogi	SHUESO	Akira Takeuchi
Shogi 5x5	SHOKIDOKI	Harm Geert Muller
Surakarta	SIA	Mark Winands

Table 1: The winning programs of the 17th Computer Olympiad.

[1] Tilburg center for Cognition and Communication (TiCC), Tilburg University, Tilburg, The Netherlands. Email:{H.J.vdnHerik,A.Plaat,J.W.Hellemons}@uvt.nl.
[2] JAIST, Kanazawa, Japan. Email: iida@jaist.ac.jp

INVADER DEFENDS AMAZONS TITLE

Richard J. Lorentz[1]
Northridge, California, USA

The ICGA 2013 Computer Game Tournaments was held this year in Yokohama, Japan, August 12th – 18th, 2013 in conjunction with The World Computer Chess Championship and the Computer and Games 2013 Conference. Last-minutelogistic difficulties were overcome by relocation to the convenient and very comfortable Hiyoshi Campus of Keio University. Indeed, all events ran quite smoothly and an enjoyable time was had by all.

This year there were six competitors in the Amazons event (see Table 1), two more than were seen in the previous four competitions. 8QP and INVADER are long time participants (competing for the first time in 2000 and 2001, respectively) and have battled each other for the gold medal for the last nine years. ARROW 2 was back for its third consecutive appearance but is still not performing up to expectations. FORTRESS returned for the second time and seems to have made good progress over last year. The two new entries this year were LONG SHOT and EXPLORER. EXPLORER managed a nice win over ARROW 2 to avoid losing all of its games while LONG SHOT did very well for a newcomer, finishing in the top half.

Program	Author	Country
ARROW 2	Martin Müller	Canada
EXPLORER	Zhou Ke	China
FORTRESS	Andi Zhang	China
INVADER	Richard Lorentz	U.S.A.
LONG SHOT	Marcin Malec	U.S.A.
8QP	Johan de Koning	The Netherlands

Table 1: The participants.

The tournament was a double round robin, thus allowing each program a chance to play with both colors against its opponents. As has been the case in past tournaments, playing first did not seem to provide any advantage. The overall score had White winning just two more games than Black.

The top three finishers dominated the bottom three by winning all of their games, essentially creating two smaller tournaments within the larger. The contest in the bottom half was quite tight, as can be seen in Table 2 below. One more win by EXPLORER could have created a three-way tie. Meanwhile, the scores in the top half, though seemingly indicating clear demarcations, was actually closer than it appears. There were a number of close, hard-fought games and it is easy to imagine quite different results if a few key moves in a couple of important positions had been different.

Place	Program	INV	8QP	LSH	FOR	ARR	EXP	Score
1	INVADER (INV)	x	2	2	2	2	2	10
2	8 QUEENS PROBLEM (8QP)	0	x	2	2	2	2	8
3	LONG SHOT (LSH)	0	0	x	2	2	2	6
4	FORTRESS (FOR)	0	0	0	x	1	2	3
5	ARROW 2 (ARR)	0	0	0	1	x	1	2
6	EXPLORER (EXP)	0	0	0	0	1	x	1

Table 2: Results and final standings.

The battles between INVADER and 8QP have always been exciting and this year was no exception. Scheduling of the matches was such that all other games were played before INVADER and 8QP met. Since both were undefeated at this point the gold medal winner would be decided by these games. The first game ended up being rather uninteresting. INVADER and 8QP played according to their usual styles, that is, INVADER tried to

1 Department of Computer Science, California State University, Northridge, CA 91330-8281, USA. Email: lorentz@csun.edu

trap one of 8QP's pieces while 8QP mapped out huge swathes of territory as compensation. But INVADER took a slight advantage early on and 8QP was never able to catch up.

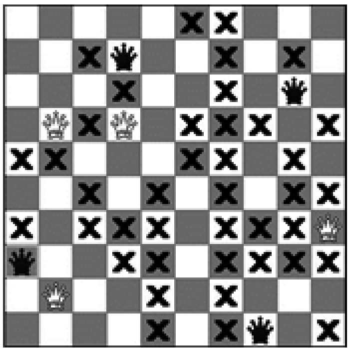


Diagram 1. 8QP vs. INVADER, Game #2 after move 40.

The next game decided whether INVADER would get the gold medal or a playoff would be necessary. Again, the game began predictably, with INVADER trying to trap and 8QP trying for territory. The results of these strategies can still be seen as late as move 40, shown in Diagram 1. INVADER, playing black, has trapped one of 8QP's white pieces on j4 but 8QP has considerable compensation. On the left side of the board INVADER has the advantage of a three against two battle, something that programs are notoriously bad at evaluating. (They tend to underestimate grossly the advantage to the side with the extra piece.) Also, on the right side, though White (8QP) has a piece that is completely surrounded, Black only has 15 points of territory for its two surrounded pieces. Since the rule of thumb is that each piece should control about 10 points of territory this effectively means that White only has to recoup five points on the left side where the program has the advantage. Strangely, though, for the last 20 or so moves INVADER has been evaluating this position as very positive, giving win rates of over 90% (being an MCTS based program), while 8QP has more correctly been showing only a very slight advantage for Black. In fact, the win rate reported by INVADER after **40. d3-a3(d3)**, leading to the position in Diagram 1, was 93%.

But then panic struck the INVADER camp. White followed up with **41. d7-e8(d7)** and suddenly Black's evaluations are showing win rates around 75%. What has INVADER suddenly seen that he wasn't seeing before? How could the win rate drop by nearly 20% after just one move? Historically, such a sudden drop in win rate presages more of the same as the game progresses. Gripped with fear, the INVADER team could only watch in anguish as the game played out and simply hope that things weren't as dire as they now seemed.

As it turns out, they were not. INVADER maintained win rates around 75% for the next few moves and then eventually consolidated its position to squeak out a win by two points thus securing the gold medal, the fifth in a row for INVADER.

I cannot overstate how much fun and how exciting the Computer Game Tournaments are, especially when they are as well organized and well-run as they always seem to be. I applaud the ICGA for the continuing and tireless efforts they make for the benefit of the computer games community.

MC-LOA WINS LINES OF ACTION TOURNAMENT

Marc Lanctot and Mark Winands[1]
Maastricht, the Netherlands

Lines of Action (LOA) was not present at the 2010 and 2011 tournaments. The return this year has marked the eighth time that LOA was played at this event. There were two participants, MC-LOA and DEEP NIKITA. This year marked a fundamental change in the search paradigm, a shift to Monte-Carlo Tree Search (MCTS). Mark Winands participated for the first time with a MCTS version of his old engine MIA, called MC-LOA. The previous αβ-based program MIA won the last four LOA tournaments, receiving two silver medals and one bronze medal before. Andrew Lin participated for the first time with his αβ program DEEP NIKITA.

The LOA tournament was played on Thursday, August 15th, 2013. All six games led to convincing wins by MC-LOA. The final standings of the LOA tournament are given in Table 1.

Rank	Program	Author	Country	Points
1	MC-LOA	Mark Winands	The Netherlands	6
2	DEEP NIKITA	Andrew Lin	USA	0

Table 1: The final standings of the LOA tournament.

[1] Games and AI Group, Department of Knowledge Engineering, Faculty of Humanities and Sciences, Maastricht University, P.O. Box 616, 6200 MD Maastricht, The Netherlands. Email: m.winands@maastrichtuniversity.nl.

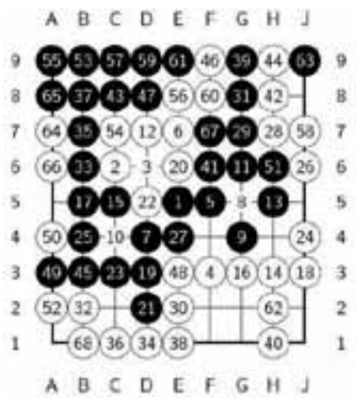
MC-LOA took the gold medal, and DEEPNIKITA received the silver medal. DEEPNIKITA is an αβ engine that was able to search to 10-12 plies, but was nonetheless unable to compete against a MC-LOA, a parallel Monte Carlo MCTS engine with sophisticated heuristic knowledge. Ongoing discussion with DEEPNIKITA's author, Andrew Lin, will hopefully encourage another LOA tournament at the next event.

GOLOIS WINS PHANTOM GO TOURNAMENT

Tristan Cazenave[1]*, Shi-Jim Yen*[2], and *Cheng-Wei Chou*[2]
Paris, France Hualien, Taiwan

There were two participants for Phantom Go at the 2013 Computer Olympiad in Yokohama. Phantom Go is a variation on Go where you do not see the opponent moves, it is only when you play an illegal move or when stones are captured that you discover the opponent position. The two programs that participated were NDHUPHANTOMGO by Cheng-Wei Chou and Shi-Jim Yen, from NDHU, Taiwan, and GOLOIS by Tristan Cazenave from University Paris-Dauphine, France.

NDHUPHANTOMGO uses MCTS, pattern matching and rules. Patterns are matched on each empty point and rules help finding the strings when stones are captured and avoiding playing on the first line in early stage of the game. GOLOIS uses raw Monte Carlo to choose its moves. GOLOIS won all of its eight games against NDHUPHANTOMGO. GOLOIS was awarded the gold medal and NDHUPHANTOMGO the silver medal.



Game two: GOLOIS is black.

The prize ceremony. Left to right: Shi-Jim Yen, Tristan Cazenave, and Jaap van den Herik

In most of the games, GOLOIS used a splitting strategy. It consists in dividing the board in two with a wall of stones and claiming one of the two sides by preventing any life for the opponent in it.

[1] LAMSADE, Universit Paris-Dauphine, Paris, France. Email: cazenave@lamsade.dauphine.fr
[2] NDHU, Hualien, Taiwan, email: {sjyen@mail.ndhu.edu.tw; kapakapa@gmail.com]

SIA WINS SURAKARTA TOURNAMENT

Marc Lanctot and Mark Winands[1]
Maastricht, The Netherlands

This year was the fifth time that Surakarta was played at the Computer Olympiad. Four programs competed in the 2013 tournament: SIA by Mark Winands (The Netherlands), BITSKT by Rui Li (China), DEEP NIKITA by Andrew Lin (USA), and V&S SURAKARTA by Ke Zhou (China). Mark Winands participated with SIA for the fourth time. It won the Surakarta tournament in 2007, 2008, and 2010. SIA still used the αβ search engine of MIA that won the LOA tournaments in 2003, 2004, 2006, and 2009. Besides the standard αβ-enhancements, the program applied multi-cut forward pruning and realization probability search.

The Surakarta tournament started on Wednesday, August 14th, 2013. The programs played two games against each other. SIA won convincingly all six of its games. The final standings of the Surakarta tournament are given in Table 1.

Place	Program	SIA	BITSKT	DEEPNIKITA	V&S SURAKARTA	Points
1st	SIA	x	2-0	2-0	2-0	6
2nd	BITSKT	0-2	x	1.5-0.5	1.5-0.5	3
3rd	DEEP NIKITA	0-2	0.5-1.5	x	2–0	2.5
4th	V&S SURAKARTA	0-2	0.5-1.5	0–2	x	0.5

Table 1: The final standings of the Surakarta tournament.

In the end SIA took the gold medal, BITSKT was awarded with the silver medal, and DEEP NIKITA received the bronze medal.

CALENDAR FOR EVENTS 2014

August 2014
ECAI Computer Games Workshop 2014, Prague, Czech Republic. More information: http://www.lamsade.dauphine.fr/~cazenave/cgw2014/cgw2014.html

August 19-21, 2014
The 2014 Chinese Computer Games Tournament, Chengdu, Sichuan Province, China. More information: Prof. Xinhe Xu, email: xuxinhe@mail.neu.edu.cn.

October 18-23, 2014
18th World Computer-Bridge Championship to be held at the 14th World Bridge Series, Sanya, Hainan, China. More information: http://www.worldbridge.org/2014-world-bridge-series.aspx

1 Games and AI Group, Department of Knowledge Engineering, Faculty of Humanities and Sciences, Maastricht University, P.O. Box 616, 6200 MD Maastricht, The Netherlands. Email: m.winands@maastrichtuniversity.nl.

THE 17TH WORLD COMPUTER-BRIDGE CHAMPIONSHIP

Alvin Levy[1]
Commack, USA

The ACBL/WBF World Computer-Bridge Championship is held annually at a major human championship. This year's event was held on September 23-28, 2013 at the World Bridge Federation's 41st World Teams Championships in Bali Indonesia. Six of the best robots were entered, including: the two top past winners, defending champion Jack (The Netherlands) and WBridge5 (France), past champions Shark Bridge (Denmark) and Bridge Baron (USA), and many time runner-ups Q-Plus Bridge (Germany) and Micro Bridge (Japan).

The format was a 48-board round robin with the two top finishers playing for the Gold medal in a 64-board KO match. The Conditions of Contest call for a semifinal stage when there are seven or more entries, but only a final KO with six or fewer teams. Twice before, in 2001 and 2005, were there as few as six robot teams entered. The greatest number of entries was ten, in 2009.

The contestants all used the same computers, 2.9 GHz Intel Core i5 desktop PCs under Windows 7 OS.

The round robin ended with WBridge5 (69.45), Jack (60.11), Q-Plus Bridge (54.81), Micro Bridge (48.03), Shark Bridge (39.13), and Bridge Baron (28.47).

Board 37 from the last round robin match had the theme ... bid one more!

Dealer: North Vul: N/S	♠ K Q 10 2 ♥ Q 10 9 7 ♦ 8 2 ♣ 10 9 2	
♠ 9 8 5 3 ♥ A ♦ Q 6 5 ♣ K Q 5 4 3	N W E S	♠ A J 7 6 4 ♥ J 5 ♦ 9 4 ♣ A J 8 7
	♠ — ♥ K 8 6 4 3 2 ♦ A K J 10 7 3 ♣ 6	

West	North	East	South
Micro Bridge	*Jack*	*Micro Bridge*	*Jack*
	Pass	1 ♠	3 ♣[1]
4 ♥	Pass	4 ♠	5 ♥ **
5 ♠	Dbl	Pass	Pass
Pass			

[1] hearts and diamonds

The play started ♦K, ♦A, ♦x, ruff and over-ruff. Declarer leads a heart to the ace and leads a trump, covering the 10 with the Jack and holds the trump losers to one for down one in 5♠x. E/W -100.

At the other table,

West	North	East	South
Jack	*Micro Bridge*	*Jack*	*Micro Bridge*
	Pass	1 ♠	2 ♠[1]
4 ♠	Pass	Pass	Pass

[1] hearts and a minor

[1] Commack, USA. Email: allevy@aol.com

The play started the same way ♦K, ♦A, ♦x, ruff and over-ruff and again declarer had no trouble holding the trump losers to one, making 4♠. E/W +420. 11 IMPs to JACK.

In another match:

West	North	East	South
SHARK BRIDGE	*Q-PLUS BRIDGE*	*SHARK BRIDGE*	*Q-PLUS BRIDGE*
	Pass	1 ♠	3 ♣[1]
4 ♠	Pass	Pass	5 ♥ **
Dbl	Pass	Pass	Pass

[1] hearts and diamonds
Making 5, N/S +850

West	North	East	South
Q-PLUS BRIDGE	*SHARK BRIDGE*	*Q-PLUS BRIDGE*	*SHARK BRIDGE*
	Pass	Pass	4 ♥
Dbl	Pass	4 ♠	Pass
Pass	Pass		

The play started ♦K, ♦A, ♦x, ruff and over-ruff and again declarer had no trouble making 4♠, E/W +420.
15 IMPs to Q-PLUS BRIDGE.

In the other match:

West	North	East	South
BRIDGE BARON	*WBridge5*	*BRIDGE BARON*	*WBridge5*
	Pass	1 ♠	3 ♣[1]
4 ♠	Pass	Pass	5 ♦ **
Dbl	5 ♥	Dbl	Pass
Pass	Pass		

[1] hearts and diamonds
Made 5, N/S +850

West	North	East	South
WBridge5	*BRIDGE BARON*	*WBridge5*	*BRIDGE BARON*
	Pass	1 ♠	2 ♥
4 ♥[1]	Pass	4 ♠	Pass
Pass	Dbl	Pass	Pass
Pass			

[1] strong spade raise with heart control.

The play started ♦K, ♦A and a switch to ♥4. South recognized that there was no need to ruff out dummy's diamond winner. Now declarer led a spade to the JACK, and had to loss two trumps for down one, E/W -100. 13 IMPs to WBridge5
** the developers comment on their robot's choice of 5♦/5♥.

Hans Kuijf, developer of JACK, comments "Robots are certainly less partner oriented than humans. Best bridge is probably somewhere in the middle. Humans include partner too much and robots too little. The reason for JACK's 5♥ bid is that it is rule based and not based on simulations. If partner is able to make the right decision based on a number of sample hands, then JACK invites partner. In this case 5♦ would tend to show longer diamonds than hearts. Humans in the South seat, however, will certainly bid 5♣ or 4NT".

Hans Leber, developer of Q-PLUS BRIDGE comments. "It would be nice to say that 5♥ must show 6-6, because with 6-5 it would not bid 3♣ (but possibly with 5-6 it would). This is how Q-PLUS BRIDGE is programmed, but after 4 ♠ the robot sees the advantage of bidding on and prefers the major over the minor without *thinking*, i.e., purely rule based. In this situation I would have expected the robot to have make a simulation, but it did not because it did not consider it a choice between 5♦ and 5♥." Hans Leber considers this a software error. "If it had run a simulation (which I did after the play) it would have selected 5♦. So 5♥ is an error which turned out lucky. Over South's 5♦ North would pass if West passed, but run to 5♥ if 5♦ was doubled."

Yves Costel, developer of WBride5 comments "WBridge5 has a rule to add one to the length of a long suit with AKQ or AQJT or AKJT. In that case diamonds are considered longer than hearts and WBridge5 bids 5♦."

Board 10 of the final round robin started the same way at all six tables, with East dealer, the bidding started 1♥ - 3 ♠ - Pass - Pass. At five tables East reopened with a Dbl.

Dealer: East Vul: Both	♠ 8 6 ♥ A 9 7 ♦ Q J 5 3 ♣ 9 4 3 2	
♠ Q 9 2 ♥ 10 6 ♦ 8 6 4 2 ♣ K Q 10 8	N W E S	♠ K ♥ K Q 8 4 3 ♦ A 10 9 ♣ A J 6 5
	♠ A J 10 7 5 4 3 ♥ J 5 2 ♦ K 7 ♣ 7	

West	North	East	South
MICRO BRIDGE	*JACK*	*MICRO BRIDGE*	*JACK*
		1♥	3♠
Pass	Pass	Dbl	Pass
3 NT	Pass	Pass	Pass

Opening lead ♠8, making E/W +600

In one match MICRO BRIDGE (West) bid 3NT, as shown above, and could not be stopped from taking nine tricks, while at the other table, JACK (West) bid 4♥ and went down 3 for -300 and 11 IMPs to MICRO BRIDGE. In another match SHARK BRIDGE (West) bid and made 4♣ for +130 while at the other table Q-PLUS BRIDGE (West) passed. In 3♠x SHARK BRIDGE played correctly by setting up a diamond trick for a heart discard before playing trumps and was +730 (only double dummy defense can beat 3♠) and 13 IMPs to SHARK BRIDGE. In the final match WBridge5 (East) doubled and WBridge5 (West) passed. After a heart lead, East cashed the club ace and the diamond ace on which BRIDGE BARON correctly unblocked the king (not needed in this particular layout) and was +730. At the other table, BRIDGE BARON did not balance with a double, and WBridge5 went down one in 3♠ when declarer played on trumps before diamonds. 13 IMPs to BRIDGE BARON.

For a comparison to the human play, two 16-board sessions were taken from the championship round robin play (round robin sessions 1 and 14), and used in the final 16-board session of the 2nd and 3rd robot round robin. The human competition consisted of 22 teams in each of three categories, Open (Bermuda Bowl), Women (Venice Cup) and Seniors (d'Orsi Trophy). One can compare the robot results to the human results as all the robot play records are shown and the links to the human records are given on the official website, www.computerbridge.com

Round 14 of the human round robin was used in the third round of the robot round robin. On board 12, 7♥ was the final contract at three of the six robot tables and 6♥ was the contract at the other three tables. In one match JACK picked up 13 IMPs against SHARK BRIDGE.

Dealer: West ♠ A 8 5 3
Vul: N/S ♥ A K J 5 2
♦ A
♣ A J 4

West		East
♠ 10 9 6 2	N	♠ Q J 7 4
♥ 7	W E	♥ 3
♦ K Q J 10 7 6	S	♦ 9 5 4 2
♣ 6 5		♣ 10 8 7 3

♠ K
♥ Q 10 9 8 6 4
♦ 8 3
♣ K Q 9 2

West	North	East	South
SHARK BRIDGE	*JACK*	*SHARK BRIDGE*	*JACK*
2 ♦	Dbl	4 ♦	4 ♥
5 ♦	7 ♥	Pass	Pass
Pass			

West	North	East	South
JACK	*SHARK BRIDGE*	*JACK*	*SHARK BRIDGE*
3 ♦	Dbl	Pass	4 ♥
Pass	5 ♥	Pass	6 ♥
Pass	Pass	Pass	

In another match BRIDGE BARON picked up 13 IMPs against MICRO BRIDGE.

West	North	East	South
Baron Baron	*MICRO BRIDGE*	*Baron Baron*	*MICRO BRIDGE*
3 ♦	Dbl	5 ♦	5 ♥
Pass	6 ♥	Pass	Pass
Pass			

West	North	East	South
MICRO BRIDGE	*Baron Baron*	*MICRO BRIDGE*	*Baron Baron*
Pass	2 ♣[1]	Pass	2 ♥
3 ♦	4 ♥	Pass	4 NT
Pass	5 ♥[2]	Pass	5 NT
Pass	6 ♣[3]	Pass	7 ♥
Pass	Pass	Pass	

[1] 2NT or game force; [2] 2 or 5 key cards; [3] no kings

In the other match WBridge5 picked up 13 IMPs against Q-PLUS BRIDGE.

West	North	East	South
Q-PLUS BRIDGE	*WBridge5*	*Q-PLUS BRIDGE*	*WBridge5*
Pass	2 ♦[1]	Pass	2 NT[2]
Pass	3 ♥	Pass	4 NT
Pass	5 NT[3]	Pass	7 ♥
Pass	Pass	Pass	

[1] game force; [2] no ace, 7+ hcp; [3] 5 key cards

West	North	East	South
WBridge5	*Q-Plus Bridge*	*WBridge5*	*Q-Plus Bridges*
Pass	2 ♣[1]	Pass	2 ♥
Pass	3 ♥	Pass	3 ♠[2]
Pass	4 NT	Pass	5 ♣[3]
Pass	6 ♥	Pass	Pass
Pass			

[1] 2NT or game force; [2]control; [3]0 or 3 key cards

In human competition 7♥ was reached 9 times in the Bermuda Bowl, 9 times in the Venice Cup (played once in 4♥), and 9 times in the d'Orsi Senior Trophy (played once in 5♥). So the robots' percentage was better (50% compared to 41%) than in the human's, albeit with a small sample.

Board 15 of the same round produced a large swing on an opening lead against 6♦.
In one match, both Q-Plus Bridge and WBridge5 bid and made 6♦.

Dealer: South
Vul: N/S

North:
♠ A J 4
♥ A 4
♦ A Q 10 9 8 5 4
♣ 2

West:
♠ 10 9 7 2
♥ Q J 8 2
♦ J 7 6
♣ 10 7

N
W E
S

East:
♠ 8 6 5
♥ K 7 6 5 3
♦ —
♣ A Q 6 5 4

South:
♠ K Q 3
♥ 10 9
♦ K 3 2
♣ K J 9 8 3

West	North	East	South
Q-Plus Bridge	*WBridge5*	*Q-Plus Bridge*	*WBridge5*
			1 ♣
Pass	2 ♦	Pass	2 NT
Pass	4 ♦	Pass	4 ♠
Pass	4 NT	Pass	5 ♦[1]
Pass	6 ♦	Pass	Pass
Pass			

[1] one key card opening lead, ♠ 8, making 6, N/S +1370

West	North	East	South
WBridge5	*Q-PLUS BRIDGE*	*WBridge5*	*Q-PLUS BRIDGE*
			1 ♣
Pass	2 ♦	Pass	3 ♦
Pass	4 ♣[1]	Pass	4 ♠[2]
Pass	4 NT	Pass	5 ♦[3]
Pass	6 ♦	Pass	Pass
Pass			

opening lead, ♣ A, making 6, N/S +1370

The other time 6♦ was bid.

West	North	East	South
SHARK BRIDGE	*JACK*	*SHARK BRIDGE*	*JACK*
			1 ♣
Pass	1 ♦	1 ♥	Pass
Pass	2 ♥	Pass	3 ♣
Pass	4 NT	Pass	5 ♣[1]
Pass	5 ♦	Pass	5 ♥[2]
Pass	6 ♦	Pass	Pass
Pass			

[1] one key card, [2] not the diamond queen. N/S were on the same wavelength, that is, not stopping in 5♦.
opening lead, ♣ A, making 6, N/S +1370

At the other table SHARK BRIDGE made +660 in 3NT. 12 IMPs to JACK.

At the final match, the contracts were 3NT, +660 and 5♦, +620, with BRIDGE BARON picking up 1 IMP against MICRO BRIDGE.

In the human championships, 6♦ was reached at many tables, and for the big clubbers with South as declarer after opening 1♦. When South was declarer, West led a heart more often than not, but with North as declarer, the opposite was true. The human play (except in Daily Bulletin articles) is not revealed, so the opening lead cannot be analyzed. In 6♦ from the North side, the human defense got it right some times (25%), but the robots were 0 for 3.

In the final KO between JACK and WBridge5 there were many swing deals.

A slightly against the odds vulnerable slam by JACK produced a 13 IMP gain on Board 3 of the second quarter. Down one would have given WBridge5 13 IMPs and the crown.

Dealer: South Vul: E/W	♠ 10 5 3 ♥ J 10 ♦ 9 7 3 2 ♣ K 9 8 5	
♠ A 8 7 4 2 ♥ A Q 6 ♦ J 5 ♣ J 10 6	N W E S	♠ K Q 6 ♥ 5 3 2 ♦ A K 10 6 ♣ A Q 2
	♠ J 9 ♥ K 9 8 7 4 ♦ Q 8 4 ♣ 7 4 3	

West	North	East	South
WBridge5	*JACK*	*WBridge5*	*JACK*
			Pass
1 ♠	Pass	2 ♦	Pass
2 NT	Pass	3 ♠	Pass
4 ♠	Pass	Pass	Pass

Opening lead ♣5, making 6: E/W +680

West	North	East	South
JACK	*WBridge5*	*JACK*	*WBridge5*
			Pass
1 ♠	Pass	2 ♦	Pass
2 NT	Pass	4 ♣	Pass
4 ♠	Pass	4 NT	Pass
5 ♥[1]	Pass	5 NT	Pass
6 ♣[2]	Pass	6 ♠	Pass
Pass	Pass		

[1]2 of 5 key cards without trump queen; [2] no kings

The probability of making 6♠ is approximately 44.25%. 62.25% of the 67.8% of the times trumps are 3-2 (whenever the ♣K is onside and approximately 12.5% of the times the ♣K is offside) plus 12.5% of the 28.3% of the times trumps are 4-1. A long match can be decided by one slightly against the odds deal. Of course, the *luck* tends to balance out and without a complete analysis it can't be determined which side had the better of it.

For the complete results go to www.computerbridge.com. You will find the 17 year history of the event, along with many publications and descriptions of computer play. This year's results also offer an opportunity to compare robot play against human play, with two sets of 16 boards to compare. The complete robot play of the two sets are shown. The complete play of the final KO is also shown along with some highlights.

Setsuko Asakura: What should we have done without her in Yokohama?

THE 27TH DUTCH OPEN COMPUTER RAPID-DRAUGHTS CHAMPIONSHIP

Jan-Jaap van Horssen[1]
The Netherlands

On December 15, 2013, the 27th Dutch Open Computer Rapid-Draughts Championship was played at the office of PW Consulting in Culemborg, The Netherlands. Since 1987, this tournament is continuously organized by Leo Nagels (also participant with CERBERUS) and Jaap Bus (also referee), and sanctioned by the Royal Dutch Draughts Federation (KNDB). If we look at the list of previous champions (see Table 1), we see that in the early years several programs could dominate for a number of years, while in more recent years there seems to be more competition, but also less consistent participation of the top programs (see Nagels, Web).

1987-1989	DIOS (NL)	2002	BUGGY (FR)	2008	BOOMSTRADAM (NL)
1990-1994	TRUUS (NL)	2003	FLITS (NL)	2009	DAMY (FR)
1995-1998	FLITS (NL)	2004	DRAGON (NL)	2010	DAMAGE (NL)
1999	DIOS (NL)	2005	TORNADO (NL)	2011	MAXIMUS (NL)
2000	BUGGY (FR)	2006	DAMAGE (NL)	2012	DRAGON (NL)
2001	FLITS (NL)	2007	KINGSROW (USA)		

Table 1: Previous winners of the Dutch Open Computer Rapid Draughts Championship.

As of 2002/2003 it can be said that the best programs play at grandmaster level. At that time, FLITS and BUGGY beat two strong human grandmasters in a match. But former World Champion Guntis Valneris played FLITS to a draw in 2003 (7–7) and former World Champion Alexander Schwarzman beat MAXIMUS in 2012 (7–5) (van Horssen, 2012). Table 2 lists this year's participants, including the reigning champion DRAGON and two former champions, TORNADO and MAXIMUS. Absentees that are still active and might have a chance to win the tournament are KINGSROW (USA), DAMAGE (NL), and DAMY (FR).

Program	Author(s) + Origin	Since	Hardware	Search Threads	Engine Language	Endgame DB (# pieces)
CERBERUS	Leo Nagels (NL)	1985	Laptop 64-bit	1	Pascal	5 (6 part.)
DAM 2.2	Harm Jetten (NL)	1988	Intel i3-2100 2 cores @ 3.1 GHz	1	C	6
DRAGON	Michel Grimminck (NL)	1996	Intel i7-980 6 cores @ 3.3 GHz	6	C	8
DREAM	Said Koudache (FR)	1992	Intel i5	1	C++	3
GWD	Klaas Bor & Gijsbert Wiesenekker (NL)	1991	Intel i7-3770 4 cores @ 3.4 GHz	2 processes	C	7
MAXIMUS	Jan-Jaap van Horssen (NL)	2008	Intel i7-3930K 6 cores @ 3.2 GHz	12 hyperthreads	Java (HotSpot VM)	6
SJENDE BLYN	Jelle Wiersma (NL)	1987	Intel i7-4930K 6 cores @ 3.4 GHz	6	C++	6
SLAGZET.COM	Maurits Meijer (NL)	2009	AMD E-300 2 cores @ 1.3 GHz	1	JavaScript (Chrome)	--
TD KING	Ton Tillemans (CH)	1988	Desktop 2 cores @ 2.4 GHz	1	C	6
TORNADO	Frank Mesander (NL)	1986	Intel i7-3630QM 4 cores @ 2.4 GHz	1	C	6

Table 2: Participants of the Dutch Open Rapid Draughts 2013

Today, the relative strength of draughts programs is mainly determined by playing numerous of automated games between different engines. Based on recent match results, DRAGON could be seen as the favourite to win. Its author, Michel Grimminck, reports a score of about 52.5% against KINGSROW, which makes it probably the strongest draughts program in the world today. However, in a single round-robin tournament surprises can always happen.

[1] Zeist, the Netherlands. Email: janjaapvanhorssen@gmail.com

In the game of international draughts there is a relatively large draw margin, which is caused by the fact that in general you need four kings to catch a single enemy king in the endgame. Using endgame databases, as most programs do, even increases this effect. This means that often the stronger programs draw against each other and have to try to make the difference by winning against the weaker programs. Still there is a significant gap between the opening book and the endgame database, so the strength of a draughts program is mainly determined by the search depth and the quality of the evaluation function. Larger endgame databases are important for endgame analysis but do not contribute much to practical playing strength (van Horssen, 2012).

The playing tempo was 75 moves in 20 minutes (rapid), followed by adjudication if necessary. DRAGON convincingly won the tournament by drawing against numbers 2-4 and winning the other six games. The first four programs did not lose one game and always drew against each other. But a number of these games could have ended differently. DRAGON reached a winning database position against SJENDE BLYN, but due to a bug in the repetition hash table it started to repeat the moves so the game was declared a draw. MAXIMUS failed to win a seemingly winning macro endgame (both sides having one or more kings and still many pieces on the board) against GWD and also against outsider CERBERUS. Macro endgames are very difficult to evaluate, and in these cases a larger endgame database might have helped. GWD showed very good defending skills in the games against DRAGON and MAXIMUS. According to Gijsbert Wiesenekker, this is because of a special algorithm (a second search process) designed to find refutations of the move to play (determined by the main search process).

Probably the best game of the tournament was the victory of DRAGON with black in the classic game against TD KING, see Diagram 1. Here TD KING played 33. 36-31?, the only mistake of the game. White's only chance for a draw was to play 48-43 and exchange with 37-31. Analysis by Jaap Bus shows that the way DRAGON won this game starting with 33…20-25! is very instructive for human players (Bus, Web). This means that even at these limited time controls the quality of the games can be very high. The game between DRAGON and MAXIMUS (see Diagram 2) was perfectly balanced. After 38. 44-40 8-12 39. 48-43 24-30 40. 40-35 3-8 41. 35x24 19x30 42. 28x19 13x24 43. 33-28 8-13 44. 38-33 30-35 45. 34-30 25x34 46. 39x8 12x3 a draw was inevitable.

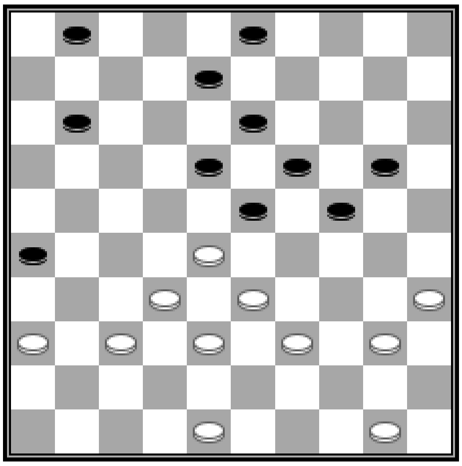


Diagram 1. TD KING–DRAGON, after 32…12-18

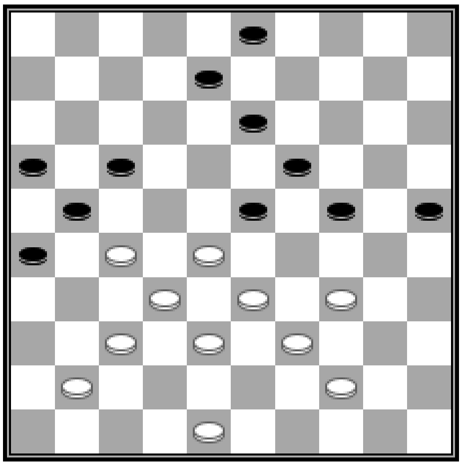


Diagram 2. DRAGON–MAXIMUS, after 37…20-25

Of the 45 games, 19 were draws (42.2%), 12 were won by White and 14 were won by Black. None of the games lasted 75 moves. TD KING experienced hardware problems before the start of the first round and received a time penalty of 5 minutes for a 30 minute delay, but managed to draw against MAXIMUS. Other than that there were no incidents. The games where a decision was reached were all won by the "stronger" program (see Table 3), either by exploiting positional mistakes early on or by outsearching the "weaker" program in the second half of the game, when concrete calculation becomes more important. Computer draughts games are rarely spoiled by tactical blunders (as human games are) so the result is mainly determined by positional play. Only in the game DREAM–TORNADO (0-2) White suddenly sacrificed a piece without a clear reason, maybe caused by a bug.

		1	2	3	4	5	6	7	8	9	10	Total	SB
1	DRAGON		1	1	1	2	2	2	2	2	2	15	112
2	MAXIMUS	1		1	1	1	2	2	1	2	2	13	98
	GWD	1	1		1	1	2	1	2	2	2	13	96
4	SJENDE BLYN	1	1	1		1	1	1	2	2	2	12	88
5	TD KING	0	1	1	1		1	2	2	1	2	11	76
6	TORNADO	0	0	0	1	1		1	2	2	2	9	50
7	DAM 2.2	0	0	1	1	0	1		1	1	2	7	44
8	CERBERUS	0	1	0	0	0	0	1		1	2	5	25
	DREAM	0	0	0	0	1	0	1	1		2	5	23
10	SLAGZET.COM	0	0	0	0	0	0	0	0	0		0	0

Table 3: Score table and final standings of the Dutch Open Rapid-Draughts 2013. See Tournament Base (Web), where all games can be viewed.

References

Bus, Jaap (Web). *End games and more*. http://www.endgamesandmore.blogspot.nl/

Horssen, J.J. van (2012). Schwarzman versus Maximus, a Man-Machine Match in International Draughts. *ICGA Journal*, Vol. 35, No. 2, pp. 106-119.

Nagels, Leo (Web). *Computerdammen*. http://home.kpn.nl/nagel580/compdam/comphist.htm

Tournament Base (Web). *Open NK Computerdraughts Commissie Computerdammen 2013*. http://toernooibase.kndb.nl/opvraag/standen.php?taal=1&kl=46&Id=3734&jr=14

The venue (Photo: Henk Stoop)

THE NSCGT-CCGC COMPUTER GAMES TOURNAMENT

Qiang Gao[14] and *Xinhe Xu*[1]

Shenyang, China

The Annual National Student Computer Games Tournament and the Chinese Computer Games Championship (NSCGT-CCGC), co-organized by the College Steering Committee of Computer Science and Technology Education(CSC-CSTE) and the Chinese Association for Artificial Intelligent (CAAI), were held simultaneously on August 13th-15th 2013 in the Harbin Engineering University of Harbin City, Heilongjiang Province, China. Professor Qidi Wu, the former Deputy Minister of Education and honorary chairman of the organizing committee and Prof. Zongli Jiang who is the deputy director of CSC-CSTE attended the opening ceremony of the tournament and also went to watch the games, which shows that the Chinese government pays great attention to the students' participation in these computer games.

The Professors Qidi Wu, Zongli Jiang, Lin Wei (the vice-president of Harbin Engineering University), and Xinhe Xu (the chairman of the organizing committee), started the tournament by touching the *laser ball* together (see picture below). There were 13 games in this NSCGT-CCGC, Chinese Chess, Go 19×19, Go 13×13, Go 9×9, Chinese Military Chess, Connect6, Dots & Boxes, Ein Stein Würfelt Nicht, Draughts, Surakarta, Amazons, Phantom Go, and NoGo. The games were played in three separate rooms. In total 168 entries (by students and non-student individuals) participated next to the individuals, all in all 260 students and teachers coming from 21 universities participated in this tournament. So, this was an unprecedented tournament in China. After the two days of fierce competition, the tournament ended successfully. The picture (below right) showed Prof. Yajie Wang (the director of the Technical Committee of Computer Games (TCCG) which belongs to the Chinese Association for Artificial Intelligence (CAAI) and vice-chairman of the organizing committee), in the middle with the winners from Shenyang Aerospace University.

Starting the Tournament.

The venue of games tournament.

Prof. Yajie Wang and winners.

[14] Computer Games Group, Northeastern University, Shenyang, 110004 P.R.China. Email: xuxinhe@ise.neu.edu.cn

No.	Game	Entries	Champion	Runner-up	Second runner-up
1	Chinese Chess	7	XQMS (Zhimin Jiang, Min Zhang)	OracleX (Zhifu Zhang)	HEU_ADE (Harbin Engineering University)
2	Go 19×19	4	Capture Hand (Minzu University of China)	HEU_RSA(GO1) (Harbin Engineering University)	HEU_ADE (Harbin Engineering University)
3	Go 13×13	3	Key Stone (Minzu University of China)	Star of HEU (Harbin Engineering University)	HRBUSTWei13 (Harbin University of Science and Technology)
4	Go 9×9	6	Miracle (Minzu University of China)	Star of HEU (GO) (Harbin Engineering University)	Mygo (Central South Un-iversity)
5	Connect6	32	Cloud (Beijing Institute of Technology)	Six or more (Beijing University of Technology)	Kha`zix6 (Shenyang University of Technology)
6	Amazons	21	TH (Shenyang Aerospace University)	Zhixin (Beijing University of Technology)	HEU_ADE (Harbin Engineering University)
7	Dots & Boxes	16	Too Young (Beijing Institute of Technology)	Tyro (Beijing Information Science and Techno-logy University)	Fighting (University of Science and Technology Beijing)
8	Surakarta	17	Vance Studio (University of Science and Technology Beijing)	Abacus2000 (Beijing Institute of Technology)	Sura (Shenyang University of Technology)
9	Phantom Go	8	neu_huanying (Northeastern University)	Sky (Shenyang Aerospace University)	KnighTeam-PG (Chongqing University of Technology)
10	NoGo	10	WTShadows (Beijing Institute of Technology)	SunSau (Shenyang Aerospace University)	Don't go (ShaoGuan University)
11	Draughts	10	GACheckers (Beijing Information Science and Technology University)	Top fighter (Shenyang University of Technology)	Grass (Beijing Institute of Technology)
12	Ein Stein Würfelt Nicht	17	iDream (University of Science and Technology Beijing)	Awareness of Future (Shenyang Aerospace University)	Star of HEU (Harbin Engineering University)
13	Chinese Military Chess	17	facing hostility on all sides (Chongqing Three Gorges University)	KnighTeam-KS (Chongqing University of Technology)	The Thunder Sword (Shenyang Aerospace University)

Table 1: The final top three of every game

The Chinese Computer Games Championship (CCGC) aims to encourage more universities and individuals to participate in computer game tournaments and Chinese Computer Games events. CCGC has been organized eight times since 2006. The number of games has been expanded to 13 from only one, Chinese Chess, in the 2006 CCGC. So, the level of Chinese computer games has kept rising. Recent results of CCGC showed that the champion of some games had been changing repeatedly and that it has been very hard to defend its championship title. The tournament was so competitive that seven games including the five well-known games Connect6, Surakarta, Ein Stein Würfelt Nicht, Chinese Military Chess, Go 19×19, have produced new champions. Table 1 lists the winners of each game.

The China Mainland entries attended numerous World Computer Olympiad competitions organized by the International Computer Games Association (ICGA) and made some achievements. NEUCHESS, developed by Northeastern University, won two Chinese Chess Gold medals (2006, 2007). Beijing Institute of Technology won the Connect6 World Championship (2009) and the Surakarta World Championship (2011). The ICGA's Computer Olympiad 2013 was held at nearly the same time as the NSCGT-CCGC. There were about 100 entries across the 20 games in the Computer Olympiad. Finally, Beijing Institute of Technology won the Ein Stein Würfelt Nicht World Championship and took second place in the Dots & Boxes World Championship and second place in the Surakarta World Championship and the Connect6 World Championship (shared second). Shenyang Aerospace University was also shared second at the Connect6 World Championship. It is thought that the China Mainland entries will achieve more prizes in future international competitions.

Since CSC-CSTE and TCCG-CAAI decided in 2011 jointly to hold the National Student Computer Games Tournament and Chinese Computer Games Championship (NSCGT-CCGC), the number of entries has grown from 91 (2011) to 168 (2013). The number indicated that the scale and influence of the competition have been greatly improved, which showed that the tournament of Computer Games is an activity that Chinese students enjoy and very much welcome.

Each tournament of computer games supports the development of general science and the technological competitions. There, theory is put into practice. The tournament atmosphere develops the students' research consciousness and innovative spirit. Besides, the tournament has some unique superiorities: (1) the board games are full of interests and competitiveness. So, they can raise the students' enthusiasm for participation and their passion for research; (2) the cost of participation is so low that an entry need only one computer; (3) the members of an entry only need to learn how to write programs; (4) the challenge is endless because of the own characteristics of the board games'; (5) as long as the communication protocols are fixed in advance, the search engine and its operation can be developed independently, so the competitions are suitable for group work; (6) justice, fairness, and equality of the competitions could be guaranteed because the rules of the competitions are transparent and there is no need of experts; (7) the computer games have better application prospects because it is easy to realize networking and industrialization of the program. Above all, the Computer Game is a competitive activity which suits students very well.

In order to ensure the healthy development of Chinese computer games' competitions, the organizing committee proposed to prohibit the use of copycat programs. In other words the entries are not permitted to use the open source or non-open source programs of others as main part of their own programs to participate in the competitions. An anti-plagiarism group was established to implement the regulations. This move obtained the desired effect and still ensured the students' enthusiasm for research and participation.

The tournament has been completed successfully, and many universities expressed the wish to hold the next tournament. After ample discussion in the organizing committee, NSCGT-CCGC will be held in Chengdu University of Technology, Sichuan Province in 2014 and in Beijing University of Technology in 2015. Moreover, new games will be added in the tournament of next year to improve the scale of the competitions and attract more universities. If possible, NSCGT-CCGC would cooperate with ICGA to organize the Computer Olympiad China Open and invite some entries overseas. It is believed that the development of Chinese computer games' events will increase with the promotion of CSC-CSTE and TCCG-CAAI.

THE 2012 ICGA JOURNAL AWARD

The Board of ICGA

In 1992, the ICGA instituted the *ICCA Journal Award*. The Award is assigned each year to a first-time author for the best article in the *ICCA Journal* in the year under consideration. The first adjudication was in 1993. In Table 1 we provide a list of *ICCA/ICGA Journal Award* winners up to 2012. Originally, the "year" ran from June to June.

The Board of ICGA is pleased to announce the recipient of the ICGA Award for the year 2011 for the best paper in this Journal by a first-time author.

Year	Award winner	Article	Published in	
1993	Peter Jansen	*KQKR: Awareness of a Fallible Opponent*	15.3	pp. 111-131
1994	Christian Donninger	*Null Move and Deep Search*	16.3	pp. 137-143
1995	Bradley Kuszmaul	*The StarTech Massively-Parallel Chess Program*	18.1	pp. 3-19
1996	Michael Buro	*Probcut: An Effective Selective Extension of the α-β Algorithm*	18.2	pp. 71-76
1997	Mark Brockington	*A Taxonomy of Parallel Game-Tree Search Algorithms*	19.3	pp. 162-175
1998	Andreas Junghanns	*Are there Practical Alternatives to Alpha-Beta?*	21.1	pp. 14-32
1999	Ernst Heinz	*Efficient Interior-Node recognition*	21.3	pp. 157-168
2000	Darse Billings	*Thoughts on RoShamBo*	23.1	pp. 3-8
2001	Thomas Thomsen	*Lambda-Search in Game Trees with an Application to Go*	23.4	pp. 203-218
2002	Ren Wu	*Fast, Memory-Efficient Retrograde Algorithms*	24.3	pp. 147-160
2003	David Fotland	*Static Eye in "The Many Faces of Go"*	25.4	pp. 203-211
2004	Haw-Ren Fang	*Checking Indefinitely in Chinese-Chess Endgames*	27.1	pp. 19-37
2005	Kohei Noshita	*Union-Connections and Straightforward Winning Strategies in Hex*	28.1	pp. 3-12
2006	Fridel Fainshtein	*A Chess Composes of Two-move Mate Problems*	29.1	pp. 3-23
2007	Gian Piero Favini	*A Program to Play Kriegspiel*	30.1	pp. 3-24
2008	Rémi Coulom	*Elo Ratings of Move Patters in the Game of Go*	30.4	pp. 198-208
2009	Nathan Sturtevant	*An Analysis of UCT in Multi-Player Games*	31.4	pp. 195-208
2010	Yoshikuni Sato	*A Shogi Program Based on Monte-Carlo Tree Search*	33.2	pp. 80-93
2011	Petr Baudiš	*Balancing MCTS by Dynamically Adjusting the Komi Value*	34.3	pp. 131-139
2012	Diogo Ferreira	*Determining the Strength of Chess Players Based on Actual Play*	35.1	pp.3-20

Table 1: The ICCA/ICGA Journal Award Winners 1993-2012.

The recipient
The Editorial Board of the ICGA Journal has recommended to assign the 2012 ICGA Journal Award for the best first-time author to Diogo Ferreira, for his article, *Determining the Strength of Chess Players Based on Actual Play*, Volume 35, No. 1, pp. 3-20.

The Board of ICGA has adopted the recommendation and congratulates Diogo Ferreira with the Award. The Award consists of three Volumes of the Journal's back issues to be chosen by the Award recipient or equivalently one of our ACG or CG proceedings. As an intrinsic part of the Award, a brief scientific biography of the recipient is published in this *ICGA Journal* (see below).

DIOGO FERREIRA: SCIENTIFIC BIOGRAPHY

Diogo R. Ferreira is professor of information systems at IST - Technical University of Lisbon, where he lectures database systems, enterprise systems integration, and business process management. He is the author of a textbook on enterprise systems integration, published by Springer in 2013. His research is mainly in the field of process mining, where the goal is to analyze the sequential behavior of processes recorded in event logs. He also has interests in chess, computer chess, and chess ratings. In 2010 he participated in a chess ratings competition organized by Jeff Sonas, where the aim was to predict the outcome of chess games based on historical data. He finished on 4th place among 250 participants. Diego tries to follow the top chess tournaments, and his all-time favorite player is Paul Keres.

MAKE SURE THE ICGA JOURNAL REACHES YOU

The *ICGA Journal* appears four times a year. In order to receive all issues of the 2013 *ICGA Journal*, you should subscribe as an ICGA member (the Triennial Meeting in Maastricht in July 2002 officially has changed the name ICCA into ICGA). The (renewal) fee is now € 40 for Europe, UK £ 25.--, or US $ 50.--. This annual fee has been agreed upon at the Triennial Meeting in Paderborn in June 1999. For subscriptions, renewals and orders of back issues, readers in **Asia, Australia, and Africa** should send their orders and payment to

ICGA
c/o Prof.dr. H. Iida
Research Unit for Computers and Games
Japan Advanced Institute of Science and Technology,
1-1, Asahidai, Nomi, Ishikawa 923-1292, Japan
Email: iida@jaist.ac.jp

Payment may be made as follows: Japanese Yens or US dollars, cash; as of January 2011 we do not accept cheques since the transition cost exceeds 23 Euro. For the readers (and others who wish to use our credit-card services, e.g., for books) payment by credit card (**Visa/Mastercard**) is possible. Send your credit-card details (Visa or Mastercard, full name on card, card number, expiry date, and address), state the amount to be paid and signed by the credit card holder to **Hiroyuki Iida, Email: iida@jaist.ac.jp, Business Fax: +81 (761) 51 1149.**

Alternatively: It is possible to pay by PayPal service via the Internet, too. All information is available at http://www.icga.org/ under ICGA Journal/Subscribe/Payment Options.

For **all other** subscriptions, renewals and orders of back issues, readers should send their orders and payment to

ICGA
c/o Prof. dr. H.J. van den Herik
Tilburg University
P.O. Box 90153, 5000 LE Tilburg / The Netherlands
Email: journal@icga.org.

Payment may be made as follows: Euros in cash, direct bank transfer from abroad: € 40. Credit-card payments should go to Hiroyuki Iida (see above). Though it is primitive, valid banknotes in UK pounds, US dollars, or Euros are still the most effective way of paying your dues neatly and safely. Also it is the cheapest way to transfer your dues free of all charges to the ICGA. Of course, postal regulations require that banknotes are sent by *registered* mail.

For the convenience of readers in countries allowing *direct transfer* to Dutch banks, payment may be made to the ING Bank (NEW BANKACCOUNT), the Netherlands, IBAN code: NL34INGB0003988921, BIC code: INGBNL2A, Tilburg, the Netherlands. While being acceptable, this method imposes charges on the ICGA and payment should be increased by Euro 7 to compensate for transfers between countries outside the European Community. Please include a correct statement of your mailing address. (We keep receiving payments without addresses!) Should your mailing label be erroneous or when changing residence, please return an amended label to Prof.dr. Iida or Prof.dr. Van den Herik as the case may require.

While stocks last, back issues of the *ICCA Journal* / *ICGA Journal* as from Vol. 6, No. 3 (August 1983) up to Vol. 35, No. 4 (2012) are still available. At this moment we offer back issues and some ICGA proceedings and books at a special price. For more information see http://www.icga.org. We can provide the pdf files from Vol. 1, No. 1 as well. Information at Johanna Hellemons: info@icga.org.

Institutional Membership is € 160 per year and includes six copies of each issue of the Journal. Library subscriptions are € 80 per year. A replacement copy for libraries is rated at € 20 each. For the American countries the amounts are US $ 190, US $ 95, and US $ 25 respectively. The currency differences are due to the differences in mailing costs.

Other enquiries should be directed to the Secretary/Treasurer, Prof.dr. Hiroyuki Iida, at the address above.

ADDRESSES OF AUTHORS

Tristan Cazenave
LAMSADE, Univ. Paris-Dauphine,
Paris / France

Bo-Nian Chen, Hung-Jui Chang, and
Tsan-sheng Hsu
Institute of Information Science, Academia Sinica,
Taipei / Taiwan

Jr-Chang Chen
Dept. of Applied Mathematics, Chung Yuan
Christian University, Taoyuan / Taiwan

Qiang Gao and Xinhe Xu
Computer Games Group, Northeastern University,
Shenyang, 110004 / P.R. China

Guy M^{c}C. Haworth
University of Reading, Berkshire / UK

Jan-Jaap van Horssen
Zeist / the Netherlands

Shun-Chin Hsu
Department of Information Management, Chang
Jung Christian University, Tainan / Taiwan.

Al Levy
Commack / USA

Richard Lorentz
Dept. of Computer Science, California State
University, Northridge, CA 91330-8281 / USA

Mehdi Mhalla and Frédéric Prost
Université de Grenoble - LIG, B.P. 53 - 38041
Grenoble Cedex 09 / France

Karsten Müller
Hamburg / Germany

Wojciech Wieczorek, Rafal Skinderowicz, Jan
Kozak, Przemyslaw Juszczuk, and Arkadiusz
Nowakowski
Institute of Computer Science, University of Silesia /
Poland

Marc Lanctot and Mark Winands
Games and AI group, Maastricht University,
Maastricht / the Netherlands

Kristian Spoerer and Simon Viennot
JAIST, Kanazawa / Japan

Shi-Jim Yen and Cheng-Wei Chou
NDHU, Hualien / Taiwan

The addresses of authors not mentioned above will be found elsewhere in this issue.

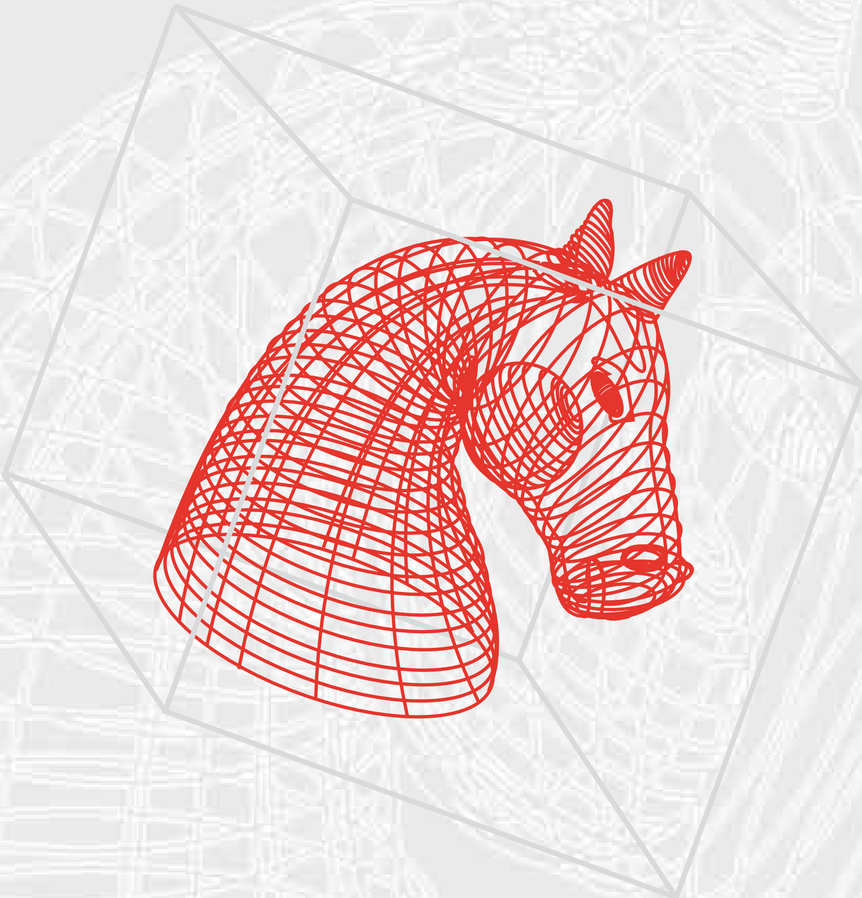
The deadline for copy for the next issue is February 1, 2014.

Please submit specific contributions to the appropriate Section Editors (see p. 233) and general contributions and correspondence to:

Prof. dr. H.J. van den Herik
Tilburg University
TiCC

Correspondence address:	P.O. Box 90153 5000 LE Tilburg / The Netherlands
Residential address:	Warandelaan 2, Building D (Dante), Room 3.46 5037 AB Tilburg / The Netherlands
Telephone:	+31 13 466 8118
Email:	journal@icga.org
Fax:	+31 13 466 2892

Please address all material to be reviewed, digested or abstracted, to Dap Hartmann, Delft University of Technology, Faculty of Technology, Policy and Management, P.O. Box 5015, 2600 GA Delft, The Netherlands. Email: L.Hartmann@tudelft.nl.



Printed by
T.O. Offset - Gijsemberg
Maastricht