Solving Kriegspiel endings with brute force: the case of KR vs. K

Paolo Ciancarini

Gian Piero Favini

University of Bologna

12th Int. Conf. On Advances in Computer Games,

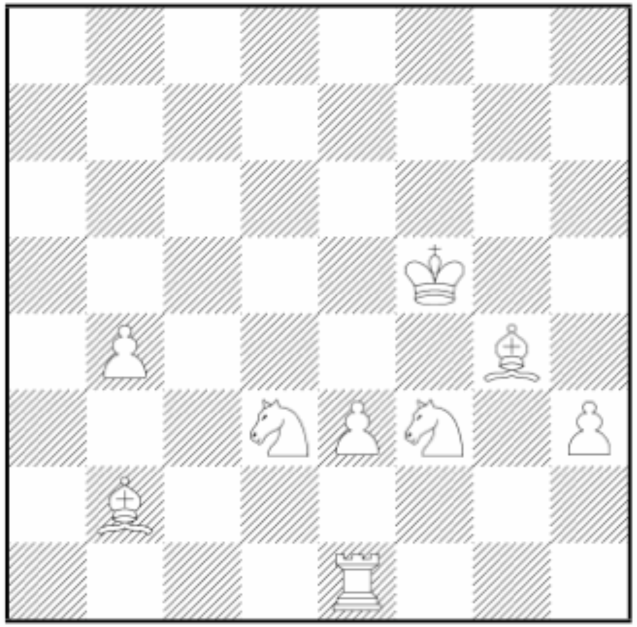
Pamplona, Spain, May 2009

The problem

- Endgame tablebases are an essential component of chess programs.
- They are built through retrograde analysis of the game tree, from the leaves up.
- Positions in the tablebase are optimal.

- Can we apply the same concepts to a game of **imperfect information**?

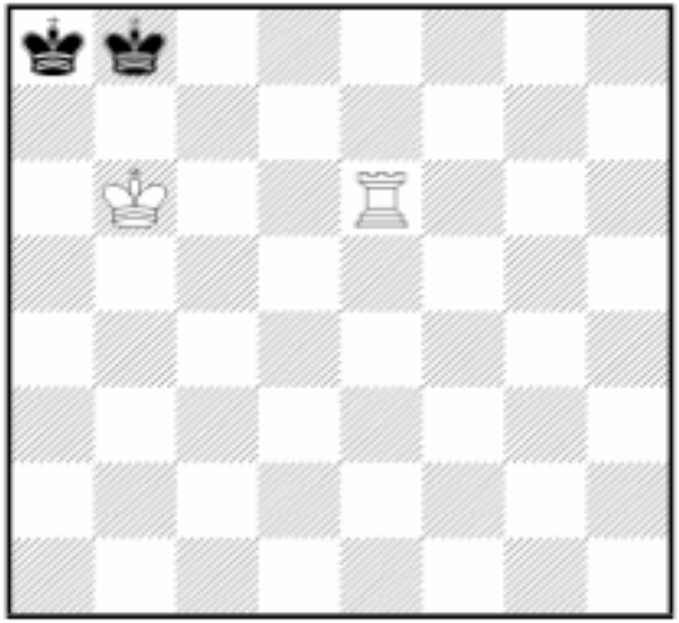
Kriegspiel



- Kriegspiel is an imperfect information version of chess.
- Players do not see their opponent's pieces; they only hear the outcome of each move from a referee.
- If a player tries to make an illegal move, he is allowed to try again.

Kriegspiel endgames

- The endgame is quite interesting in Kriegspiel, because what is enough material to mate in chess might not be sufficient in Kriegspiel.
- Mating the lone king is the main object of interest.
- We use the graphical notation on the right to represent an enemy king whose position is not perfectly known.

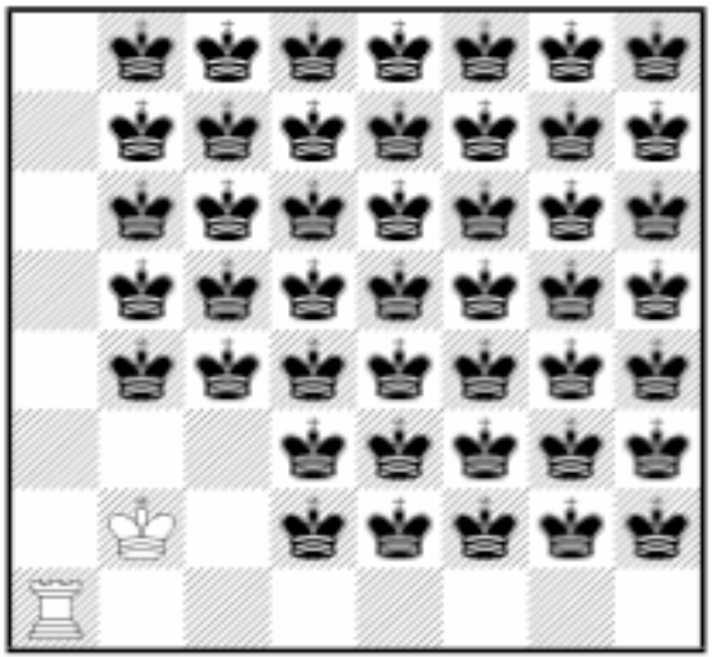


This is a mate in 1.

Notes

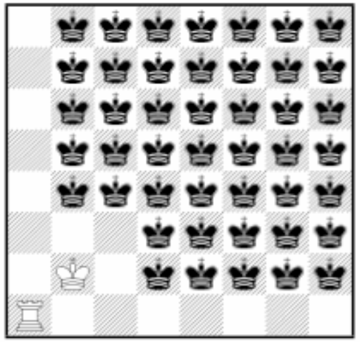
- The enemy king positions look like a "**quantum wave**" because its location is undetermined until we try a move.
- Some Kriegspiel endgames can only be won with probability approaching 1; we limit our analysis to cases where victory is certain.
- This is almost always the case in our scenario, the **KRK** endgame (King and Rook versus King).

Existing solutions (1/2)



- So far, there were two ways of solving these endgames, neither of which is fully satisfactory.
- Method #1: informal algorithms such as Boyce for KRK.
- “Reach this position and then try these moves”
- Does it always work? Is it optimal? How difficult is it to implement?

Existing solutions (2/2)



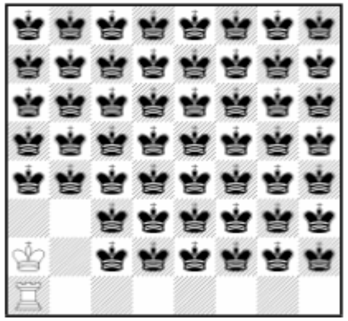
= X

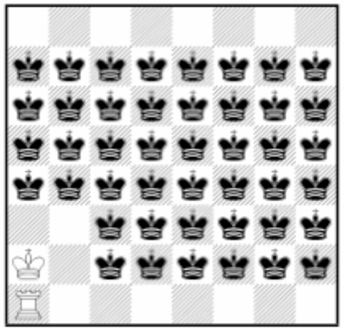
- Method #2: evaluation function.
- Each position can be evaluated according to some heuristics (such as, having kings on the edge of the board is better than in the middle). We then run a minimax-like search to decide our move.
- This performs better than an informal algorithm, but is not flawless, either (possible loops, incomplete or suboptimal functions)

Our approach

- We apply retrograde analysis to Kriegspiel endgames.
- Our algorithm takes a list of positions that can be won in up to X moves, and generates the list of positions that can be won in (X+1) moves.
- The goal is to have a full list of won positions, together with the “optimal” move.
- “Optimal” means that it leads to the fastest mate in the worst case, without making any assumptions on the opponent.

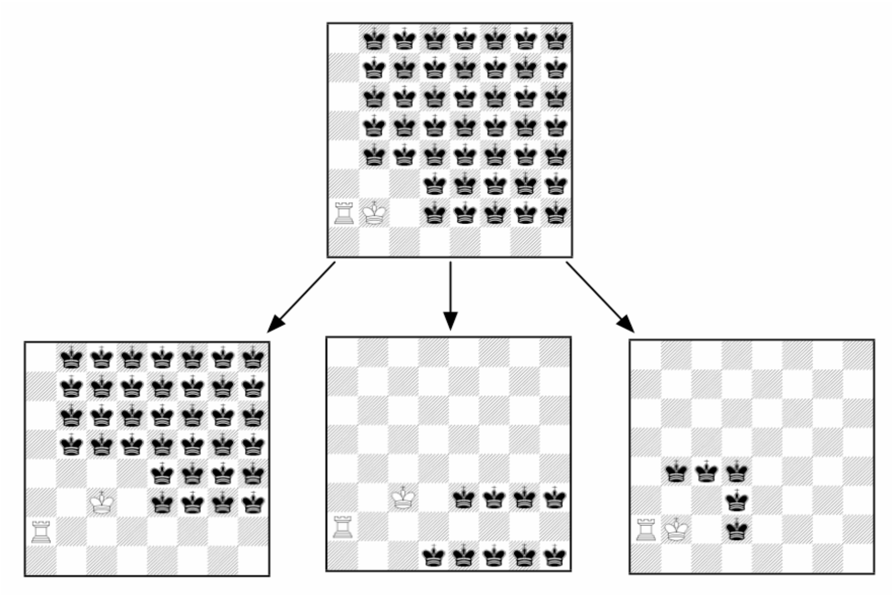
Problem size





- The Kriegspiel state space is huge. The main risk is that the list be too large to be computed and stored.
- There are up to 52 possible king squares on a KRK board.
- Even with mirroring, there are 630 ways to place White's pieces.
- There are ~ 10^{17} theoretical combinations.
- However, most of them are irrelevant as they become undistinguishable after two plies.

Board generation



- Given a move, a position can generate several new positions depending on the referee's feedback.
- In this example, Kc3 can be **silent**, **rank check** or **illegal**.
- Our algorithm performs the inverse step; starting from the three possible results, it would recover the starting position.

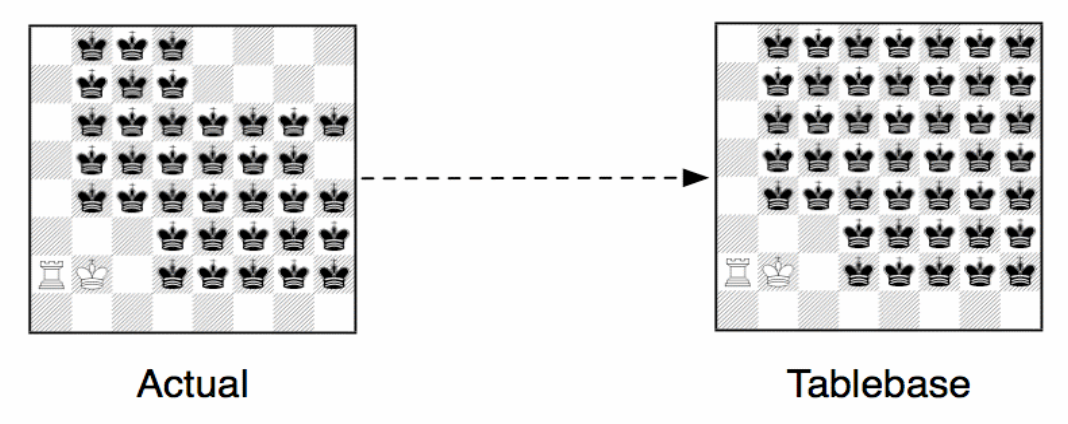
The algorithm

- It maintains a list of “active” positions.
- For all moves and White piece layouts, all compatible positions are tried as the hypothetical results of that move.
- When a new position is constructed that is not a subset of one already in the list, it is added to the list.
- When no more positions are found, the algorithm proceeds to the next depth level.
- As we try all combinations, we will necessarily find the optimal solutions.

Optimizations

- As each position is tried for each referee message, this algorithm is exponential in the number of referee messages.
- Optimization is extremely important.
- Most optimizations consist of dropping unnecessary positions and duplicates, or ignoring positions that are clearly not going to add anything.

Querying the tablebase

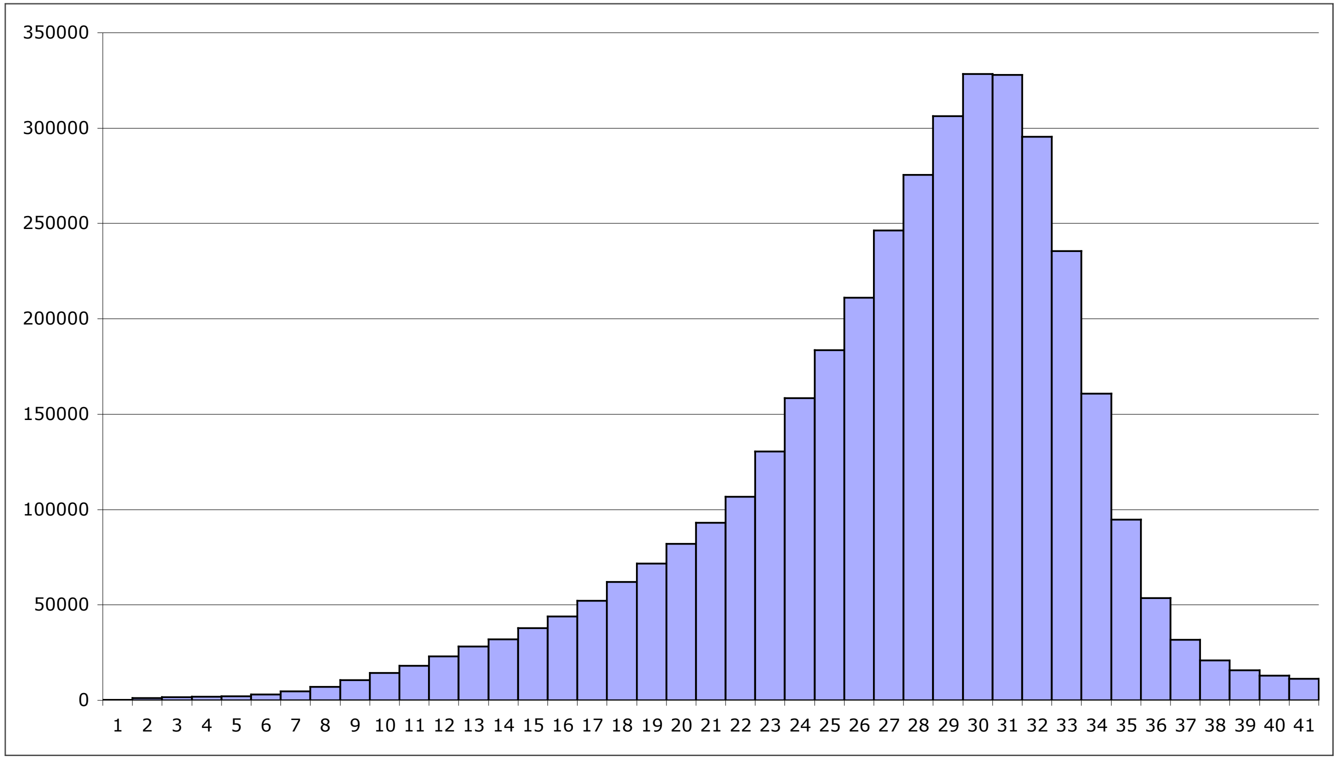


- If the current position is not in the tablebase, we return its superset with the shortest distance from mate.
- If no such superset exists, it means there is no forced mate from this position.

KRK Statistics

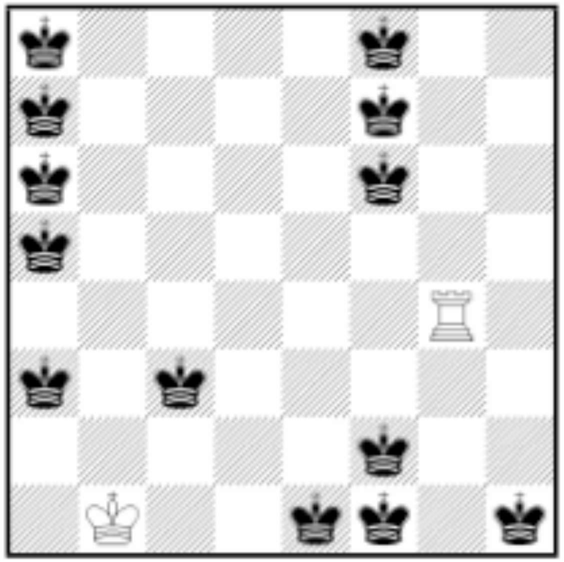
- Computed in 10 days (3 days with optimized algorithm) on a single computer.
- The resulting tablebase includes slightly over 10^6 positions; only 1 in 10^{10} positions turn out to be significant.
- The longest forced mate is 41 moves, making the 50-move rule harmless.

Active positions



- Positions that are not a subset of another position: a measure of complexity.

Longest mate



- 41 moves
- It takes White 9 moves to secure the Rook: Rf4, Kc2, Rf8, Kd3, Rg8, Rh8, Rh1, Rd1, Kc2.
- Many positions in the tablebase look like riddles, but their subsets can occur in normal play.

Future research

- Can our approach be extended to cases where White wins “almost” certainly?
- KRK is actually the simplest Kriegspiel endgame. Other cases are being researched.
  - KQK (interestingly, harder to solve than KRK, due to more possible referee messages)
  - KBBK and KBNK

Question time