Representing Kriegspiel States with Metapositions

Paolo Ciancarini and Gian Piero Favini
University of Bologna, Italy

IJCAI
Hyderabad, Jan 2007

Summary

- Kriegspiel: a board game based on partial infomation
- Our approach
- Some results
- Future works

Complete or partial information in games

Board Games of complete information:

The current state of the game is fully accessible to each player
Examples: Chess, Go.

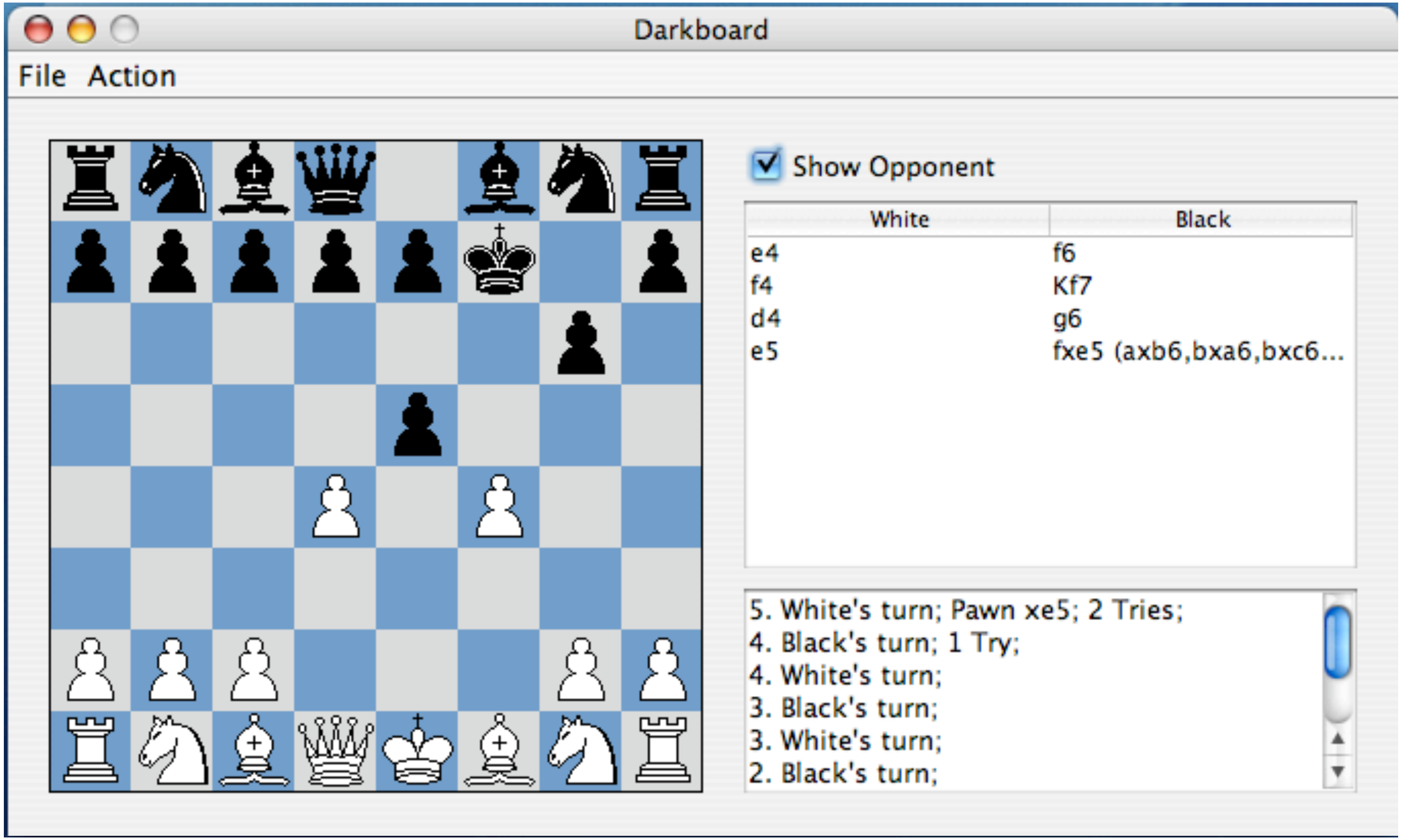
Board Games of partial information:

Players have partial (and different) knowledge about the state of the game
Examples: Battleship, Stratego, Kriegspiel

Kriegspiel

- Kriegspiel is a chess variant in which the players cannot see the opponent’s pieces. All other rules of Chess still apply
- The players never communicate directly with each other, but instead interact with a referee. Only the referee knows the full state of the game.
- Players try moves which can be either accepted or rejected by the referee; if a move is rejected another move can be tried
- The referee announces checks and captures. These messages are sent to both players.

Playing Kriegspiel



Research works on Kiegspiel

- Modeling and implementing a Kriegspiel referee program (Burger, Wetherell and others)
- Definition of algorithms for simple endings (Boyce, Ferguson, Ciancarini and others)
- Planning based on MonteCarlo Sampling
- AND-OR search of belief-state trees
- Reasoning about partially observed actions
- Algorithms for Kriegspiel variants

Why do we study Kriegspiel?

- **Complex:** extremely large belief state makes an explicit representation of it computationally intractable
- **Challenging:** currently, the best humans are still far ahead of computer players at this game
- **Convenient:** same rules as Chess: this allows for reuse of a certain amount of game theory and software

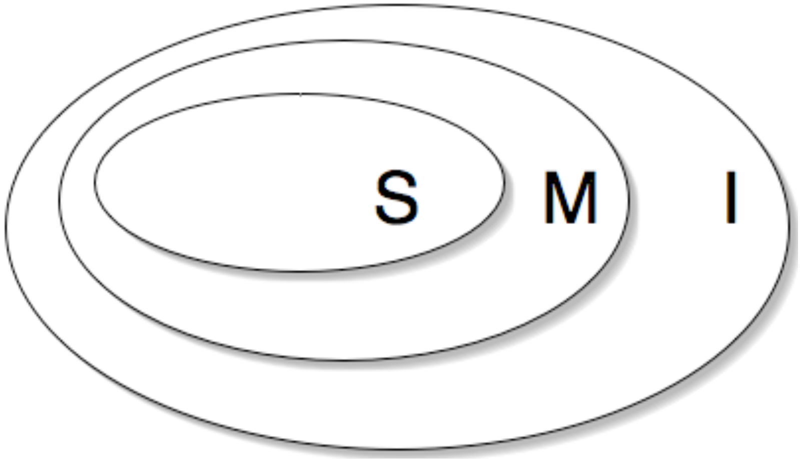
Metapositions

- Our program plays building a tree of metapositions
- A metaposition groups several game states together to provide the illusion of complete information
- The states with the same strategy space (set of moves available to the player) may be merged together and a game tree can be built
- Concept introduced in [Sakuta 2001] to deal with a Shogi equivalent of Kriegspiel, used to solve endgame positions

Metaposition

Definition: If **S** is the set of all possible game states and $\mathbf{I} \subset \mathbf{S}$ is the information set including all game states compatible with a given sequence of observations (referee's messages), a metaposition **M** is any opportunely coded subset of **S** such that $\mathbf{I} \subset \mathbf{M} \subset \mathbf{S}$

Meaning of **M** (1/2)



- A metaposition **M** is any superset of the belief state **S**, which is computationally intractable (~10^{27} states in an average Kriegspiel middlegame position).
- **M** can, however, be represented opportunely by a coding depending on the game being considered.
- Clearly, the more **M** tends to **S**, the better.

Meaning of **M** (2/2)

- It might appear counter-intuitive to approximate such a huge set as the belief state **S** with an even larger set **M**
- We claim that, for games with a particularly large **S** and small amounts of information available to the player, this approach might outperform methods based on the evaluation of subsets of **S**
- We code a Kriegspiel metaposition **M** using about 200 bytes of data, meaning that **M** is only a rough approximation of **S**

Metapositions vs Monte Carlo

- The two methods approach the problem from opposite directions; a large superset vs a small subset of the belief state.
  - Metapositions do not have to choose “good” states or fear a bias in the evaluation.
  - Metapositions will require a special evaluation function and more custom code.
  - The opponent is not assumed to follow a best defence model.
  - A metaposition can estimate its own uncertainty and use it in the evaluation function to promote a search for information.

Kriegspiel Metapositions

- Simple structure; a metaposition is a board with both allied pieces and enemy “pseudopieces” on it
- Each square has a set of boolean flags telling which types of enemy pseudopieces can be on it.
- Each square has an integer number that serves as “history”, telling how many moves ago the square was last explored (not strictly required, but coding the history of the game helps prioritize some game states).
- Other information such as castling data.

Building a metaposition tree

- A metaposition can be used to take incomplete information out of the picture. Instead of playing the original game, we can play a complete information game whose states are metapositions.
- If we can build a game tree of metapositions, we can then apply a minimax-like algorithm to evaluate moves. An evaluation function would be used that examines metapositions as a whole instead of single game states.
- Updating metapositions is the crucial step here.

Pseudomoves and Metamoves

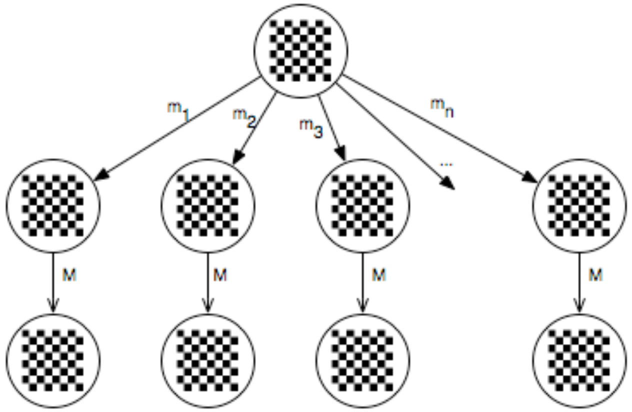
- Pseudomoves represent the player's move.
- May be rejected by the referee in Kriegspiel.
- A pseudomove and the referee's responses uniquely define the resulting metaposition.

- Metamoves represent the opponent's hidden move.
- The player does not know what the move is, so the new metaposition is uniquely defined by the referee's messages.

Approximating moves

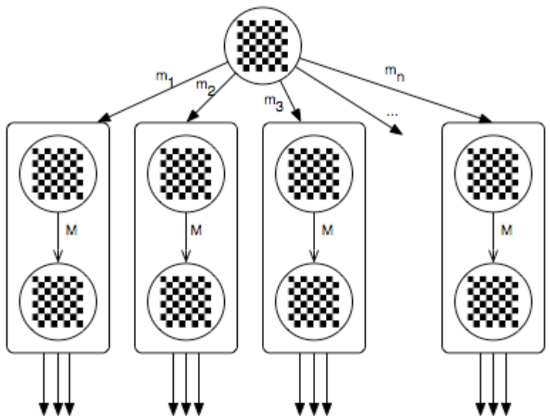
- There are too many pseudomoves (40-60 times referee responses without counting rejections) and metamoves (as many as there are possible referee messages, implying dozens of combinations) in a Kriegspiel game
- The tree needs to be pruned by predicting the referee's response, or it would become impossible to manage past 2 plies
- Hard-coded rules are used that assume the referee to be silent unless there is strong evidence to the contrary

Tree structure (1/2)



- 2-ply example.
- As many pseudomoves as there are possibly legal moves; the assumed referee response is generated with a set of rules.
- Only one metamove, generated in the same fashion.
- The agent is playing against the environment.

Tree structure (2/2)



- Since there is only one expected metamove, two plies can be merged together. Each level represents a full move.
- Instead of trying minimax, given that MIN's move does not exist anymore, the player will use maximax instead.
- Clearly, assumptions on future referee messages are being made, and it is necessary to weigh the evaluation so that deeper, less reliable nodes are also less important.

The weight α

- For each metaposition node that is not a leaf, the value is computed from its own static evaluation and that of its best child with weights α and $(1-\alpha)$, respectively.
- High values of α favour shallow nodes and immediate rewards, low values promote deeper nodes and more distant gains.
- The nature of α is two-fold: on one hand it measures the level of **risk** the program is willing to take, on the other it gauges its trust that the referee will behave as predicted

The evaluation function

- Currently uses features inferred from expert human play (a database of beyond 10.000 games played on Internet Chess Club)
- Three main components:
  - Material
  - Position
  - Information
- Material and Position have an equivalent in most evaluation functions for Chess, whereas Information is specific to Kriegspiel

Results

- Darkboard plays a decent game of Kriegspiel
- Wins 90% of games against a program choosing random moves
- Best results on Internet chess club: about 1800 Elo points, placing itself among the top 20 active players out about 200
- Winner of match-tournament against a program based on MonteCarlo sampling (Turin 2006)

Future works and conclusions

- Progress
- Strategic plans
- Opponent modeling from large set of games
- Extension of techniques to other board games based on partial information

Questions?