



Priority Level Planning in Kriegspiel

Paolo Ciancarini, Andrea Gasparro

► **To cite this version:**

Paolo Ciancarini, Andrea Gasparro. Priority Level Planning in Kriegspiel. Gerhard Goos; Juris Hartmanis; Jan van Leeuwen. 11th International Conference on Entertainment Computing (ICEC), Sep 2012, Bremen, Germany. Springer, Lecture Notes in Computer Science, LNCS-7522, pp.333-340, 2012, Entertainment Computing - ICEC 2012. .

HAL Id: hal-01556114

<https://hal.inria.fr/hal-01556114>

Submitted on 4 Jul 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Priority Level Planning in Kriegspiel

Paolo Ciancarini and Andrea Gasparro

Dipartimento di Informatica
University of Bologna, Italy

Abstract. Back in 1950, Shannon introduced planning in board games like Chess as a selective approach, where the main idea is to select specific branches of the game tree that satisfy certain conditions. He contrasted this approach with brute force Minimax-like methods, based on an exhaustive search of the game tree, that aims to select the best path inside a given search horizon. Historically, the brute force approach won hands down against planning in complex games such as Chess, as the strongest Chess programs nowadays all exploit brute force algorithms. However, planning is still interesting and even necessary in some game-playing domains, for instance based on incomplete information, where there is no way to evaluate precisely or even build the game tree. In this paper we describe a technique that produced positive results in Kriegspiel, a variant of Chess played as an incomplete information game. Our main result is the definition of an algorithm for combining MonteCarlo search with planning; we tested the algorithm on a strong Kriegspiel program based on MonteCarlo search, and obtained a clear improvement.

1 Introduction

Partial information games are a challenging domain to study and test decision making under uncertainty, that is a key concept in game theory and also a widely studied topic in many economical simulations [4]. For instance, uncertainty can be introduced in perfect information games by making invisible the opponent moves (as in a war-game referred by an arbiter [9]). This invisibility usually introduces important changes to any algorithmic playing approach: the search space size and the related game tree branching usually grow dramatically, whereas one of the key results in games theory, namely Zermelo's theorem [13], is no more valid because its assumptions do not apply. More important, these changes reflect on the effectiveness of playing algorithms which work well for the standard game [2] but must be completely redefined for its variants based on partial information.

Kriegspiel, the board game we are focusing on in this paper, is a practical example of our previous statements: removing most of the information given to a player about his opponent's moves, increases dramatically the average branching factor during middle game from 33, that is the Chess branching factor, to $33!$, that is Kriegspiel branching because a player can try its pseudolegal moves in any order. We remind that the branching factor greatly affects the performances

of Minimax and its derivative algorithms. Hence, the best Kriegspiel artificial player, as described in [6], is not based on Minimax, but on Monte Carlo Tree search instead.

This paper has the following structure: in section 2 we provide a description of Kriegspiel, introducing its rules and most important features, from the point of view of an artificial player. Section 3 contains a description of the state of the art, with special interest to Darkboard, the current world champion of Kriegspiel. Section 4 presents the main topic of this paper, namely our original planning based approach, that we applied to Darkboard to play Kriegspiel, but that can be applied to other playing domains based on partial information and with a large branching factor. In section 5 we present the results obtained by an experimental player, based on the technique we presented, matched against the current world champion program, Darkboard. Finally, in section 6 we summarize and discuss our results, concluding with the future developments we are considering.

2 Kriegspiel

Kriegspiel, or *Blind Chess*, is a chess variant invented by Michael Henry Temple at the end of the XIX century, first played and standardized at the famous London chess club "The Gambit". It can be considered a descendant of an already existent prussian war game, named *Kriegspiele*, invented at the beginning of XIX century for training Prussian army officers. Kriegspiel rules are the same as chess for what concern the goal of the game and piece movement rules, but differently from Chess, there are three main actors involved: the white player, the black player and an umpire.

Each player can see only its pieces, while from the point of view of the umpire, Kriegspiel is just a normal chess game. Being the only one informed on the real state of the board, the umpire is in charge to inform players if the moves they make are legal or they are not. For instance, a move could be illegal because it exposes the king to a check, or because the trajectory to a square is not completely free. If a move is refused by the referee, a player has to chose a different one, that is again evaluated by the umpire, and so on.

Thus, designing a program to play Kriegspiel is very different from designing a program to play Chess, due to information on the game state only partially available. Partial information grows the branching factor, while the space state remains the same of chess, that Shannon estimated at 10^{42} [11]. The reason of the larger branching factor resides in the fact that a Kriegspiel player can try its pseudo-legal moves in any order: if a move is legal then it must stay, else the player must try another move. Thus, all possible permutations of pseudo-legal moves should be enumerated to define the branching factor of Kriegspiel.

This huge increment in the branching factor reflects the fact that in each state a player does not know what is the current position of the opponent's pieces. Hence, an artificial player should somehow keep trace of all the possible configurations of the board, in each different moment of the game. This leads to the need to abstract away different positions, that are indistinguishable from

the agent point of view, into a single representation: this is the key idea behind Metapositions, first applied in Shogi [10] and successfully reused in Kriegspiel [1].

The reason why we have chosen Kriegspiel as our domain of study is that it is not yet a fully explored game, with significative differences from Phantom Go and other imperfect information games, as that the nature of Kriegspiel's uncertainty is completely dynamic, making it an interesting environment to simulate different kind of real life tasks.

3 Planning in board games: State of the Art

Recent researches in games like Phantom Go and Kriegspiel contributed to focus on a relatively unexplored category of partial information games, derived by well known perfect information games, introducing a concept of invisible moves. This area proved to be a challenging domain for artificial players based on brute force, leading to the development of strategies to handle meta states, with the aim to reduce the space state and the branching. This happened exploring metapositions in in Kriegspiel, as described in [5].

A strategy similar to the one used to play Go proved to be quite effective in Phantom Go, as described in [3], also because of the monotonous decrease of uncertainty that makes progress somehow simpler to handle, in comparison with Kriegspiel.

Until a few years ago, only a few amount of research had been devoted to Monte Carlo in games with dynamic uncertainty, with the notable exceptions of [8], which first studied planning based on MonteCarlo, and [7], which developed a more efficient approach based on a bandit algorithm called UCT (Upper Confidence bounds applied to Trees). We exploited UCT in the development of Darkboard, that is basically a program combining UCT with a node evaluation function [6].

The contributes to the study of Kriegspiel, brought by the experiments conducted developing the artificial player Darkboard, were the most significant progress in the study of the game, involving for the first time the use of metapositions in Kriegspiel, also introducing a Monte Carlo based approach to handle middle-game. Darkboard represents the state of the art in Kriegspiel playing as it is also the current Kriegspiel world champion.

4 Priority Level Planning in Kriegspiel

Kriegspiel is a hard game to play for rational agents: due to the huge branching factor mentioned above, in approaches of Shannon's "Type B", that is brute force, the ability to evaluate different states plays a key role in reducing the search space, that directly reflects on the effectiveness of brute force analysis. Unfortunately, even with the help of Metapositions, this is a very hard task for artificial players, as it involves the ability to understand the "meaning" of a position.

An approach based on planning is a more natural choice to provide an agent with a "human like" understanding of a position. However, planning was not successful in chess-playing programs, as it reduces consistently the speed of states evaluation and, consequently, the horizon of the program. This actually cuts off the possibility to base the evaluation of a state on a long range analysis.

Our approach is then based on a planning technique, meant to predict the evolution of game tree branches, basing only on factors that must be present in the present state. This requires the search of "structural factors" included in each state, that can be associated to a predictable evolution in the long run.

This technique seems to apply perfectly with Kriegspiel, because strategical elements are included in each state. Those elements can be used to form a plan, where a plan is just a sequence of transitions which fit the actual configuration of the board. An example of the strategical elements we are speaking about can be the presence of open files, if heavy pieces are still on the board, etc.

In this paper we focus on the middle-game, as it is the game phase where introducing a priority level planner does not cause any sensible change in move generation's speed. We have built a planner which during middle-game (which starts when the opening phase ends, i.e. when the program exits from its opening book, and ends when the position has no more pawns) splits move generation in two phases: first, it tries to elaborate short term plans, aiming to reach "important" goals which are enough "close", if there are not such goals reachable from the current state, it tries a long term evaluation of each *interesting* candidate transition originated from the current state.

This algorithm requires the definition of two different categories of goals:

priority goals Goals that represent a short term advantage. Examples of short term goals may be piece capture, king safety, etc.

long term goals Examples of long term goals may be placing our pieces in favorable positions, regrouping pieces in a way that makes easy for them to cooperate, etc. Each goal has a priority that represent its importance. This category includes what a chess player could call strategic or long-term goals.

The goal sets were defined according to standard chess theory. This highlights an advantage of choosing Kriegspiel as our applicative domain: a Kriegspiel planner can benefit from the well known theory of Chess, to identify key factors in a position, as long as it evaluates those factors in his own way, considering the significant differences introduced by partial information.

Our algorithm for move generation includes the following three steps:

EVALUATION The program evaluates the current board state, searching for priority goals. This task is simplified by Kriegspiel rules, since we identify priority targets thanks to umpire messages informing that a pawn try is available, that a piece has been captured or that one of the player has delivered a check.

SELECTION If a priority goal is reachable, the goal set includes only the moves that are supposed to lead to the goal, else the agent analyzes each action within two level of depths searching for long term goals.

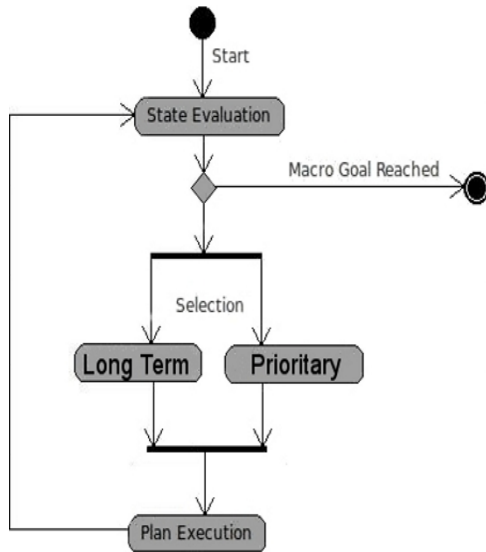


Fig. 1. Activity Diagram representing the move generation process

CONSTRUCTION The program selects the path with the highest rating. If a priority goal is detected, this unifies with the path to reach this goal.

This description highlights the fundamental importance of long term goals in move generation, as priority goals cover only a small set of possibilities. At the same time, long term goals deal with strategic factors. This technique may be promising in a domain with a clear notion of strategy, as it is the key concept used to predict the quality of a path, without the need to evaluate a large set of states.

This also means that this technique can be applied not only to Kriegspiel, but also to any strategic game with partial information and a large branching factor.

Fig 1 summarizes the process we just described.

The above algorithm contains the key idea of our approach. However, if implemented verbatim it would be highly ineffective in practice. The reason is that an agent whose move generation relies entirely upon its goal set, when no priority goal is available, is forced to define a huge number of goals to deal with each possible situation. This leads to two different problems: first, it turns out to lack flexibility to handle a complex game like Kriegspiel; second, this does not correspond to what we defined as a long term goal set.

Instead, before searching for long term goals, we execute a one step Monte Carlo analysis to filter moves that have an evaluation below some threshold. After doing so, we execute the steps described above. Nothing changes in handling

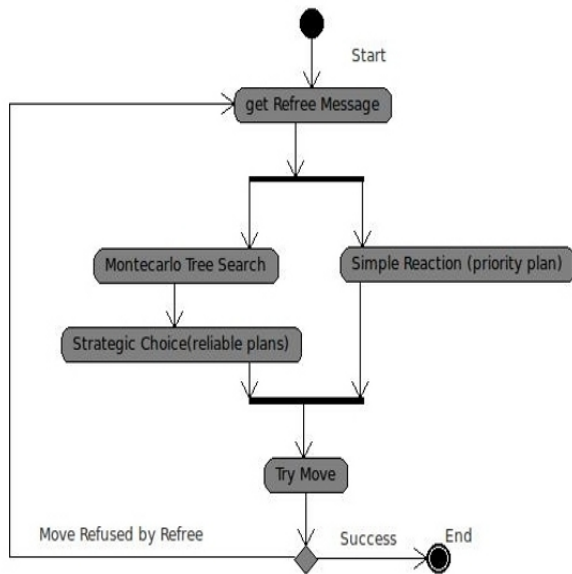


Fig. 2. Activity Diagram representing the move generation in Strategic Darkboard

short term plans, as it does not involve nothing else than an “automatic reaction” to priority goals detection. The modified behavior is shown in Fig 2.

This second approach offers two advantages. First, it does not force the agent to misrepresent the idea behind the concept of long term goals. The second advantage is that filtering the moves that the planner has to analyze reduces the computational cost dramatically, as it cuts the branching factor, while the brute force method may be limited to a very short depth. Thus, both strategies benefit one of each other.

5 Experimental results

To evaluate the result obtained introducing a priority level planning in Darkboard, we tested our enhanced player in tournaments against other programs. Those programs were:

- **World Champion** The current world champion version of Darkboard, presented in [6].
- **Random Player** A pseudo-random player, useful to evaluate the behavior of our program against an unpredictable opponent.
- **Strategic Darkboard** The World Champion program enriched with a priority level planner.

Our experiments consisted in tournaments of 20 games between Strategic Darkboard and another program. This is a well known technique, since the first Belle’s experiments by Ken Thompson [12].

The first series of experiments, consisting in 3 tournaments of 20 games each against the Random Player, confirmed that the random player is not an interesting opponent, as the average result is 85% of success for Strategic Darkboard, where all games that were not won were draws for stalemate (if the opponent king is in check and cannot move a game is draw according to chess rules).

The second series of experiments was far more interesting, consisting in 5 tournaments of 20 games each between Strategic Darkboard and the World Champion: it ended with the average success rate of 61% for Strategic Darkboard. Thus, according to the standard Elo scale, Strategic Darkboard gained a class, compared to Darkboard previous’s version.

The table summarizes the results obtained in our tests.

Table 1. Results of playing experiments

Opponent	Strategic DB	Opponent
World Champion	61%	39%
Random Player	85%	15%

A second set of experiments against human players is currently taking place on the Internet Computer Club (ICC), which hosts the largest Kriegspiel community. The partial results until now indicate a smaller improvement, but it will take some time to collect enough data to drawn some definitive conclusion.

6 Conclusions and further developments

One problem when implementing a program for playing an imperfect information game with a high branching factor is the horizon effect. Because of that, planning may be an interesting approach in such a domain, due to the fact that a planner does not analyze the totality of the states, but instead it focusses only on promising branches.

The technique we have proposed is a hybrid between planning and brute force, developed to manage two different aspect: short term game playing, that is based on brute force and somehow does not care about the branching factor, since there is a priority goal reachable inside the horizon of the agent, and long term playing, basing on the notion of strategy.

While experimental results are promising, it is important to remember that this technique relies heavily on the presence of a dense notion of strategy in the application domain, that is a key condition for limiting the depth of the searching algorithm. This limitation however is double edged, since a dense notion of strategy leads to impressing performances as shown by the experiment conducted against the world champion program of Kriegspiel.

A further development will focus on refining the long term goal set; in our experience human players, that currently outperform our Kriegspiel program, orient their game playing according to repetitive strategies: analyzing an already available collection of Kriegspiel games with a pattern recognition algorithm, may lead to identify new goals or even just to update the already existing one ratings, hopefully improving the long term view of the agent, that's the most interesting feature of this technique.

Finally, even if we obtained some interesting results, Strategic Darkboard played with a handicap: in fact, the goal set used by the program is quite limited. Future development will focus on additional strategies to populate the goal set with efficient targets, thus we expect an improvement of the playing strength.

References

1. A. Bolognesi and P. Ciancarini. Searching over Metapositions in Kriegspiel. In J. van den Herik and N. Netanyahu, editors, *4th Int. Conf. on Computer and Games*, RamatGan, Israel, 2004.
2. M. Campbell and T. Marsland. A comparison of minimax tree search algorithms. *Artificial Intelligence*, 20(4):347–367, 1983.
3. T. Cazenave. A Phantom-Go program. In T. Hsu J. van den Herik, S. Hsu and H. Donkers, editors, *Advances in Computer Games*, number 4250 in Lecture Notes in Computer Science, pages 120–125, Taipei, Taiwan, 2006. Springer.
4. P. Ciancarini, F. DallaLibera, and F. Maran. Decision Making under Uncertainty: A Rational Approach to Kriegspiel. In J. van den Herik and J. Uiterwijk, editors, *Advances in Computer Chess 8*, pages 277–298. Univ. of Rulimburg, 1997.
5. P. Ciancarini and G. Favini. Representing Kriegspiel States with Metapositions. In *Proc. 20th Int. Joint Conf. on Artificial Intelligence (IJCAI 07)*, pages 2450–2455, India, January 2007.
6. P. Ciancarini and G. Favini. Monte Carlo Tree Search in Kriegspiel. *Artificial Intelligence*, 174(11):670–684, 2010.
7. L. Kocsis and C. Szepesvari. Bandit based Monte Carlo Planning. In J. Furnkranz, T. Scheffer, and M. Spiliopoulou, editors, *Proc. 15th European Conference on Machine Learning*, number 4212 in Lecture Notes in Computer Science, pages 282–293. Springer, 2006.
8. M.Chung, M.Buro, and J.Schaeffer. Monte Carlo planning in RTS games. In G. Kendall and S. Lucas, editors, *Proc. IEEE Symposium on Computational Intelligence and Games*, pages 117–124, Colchester, Essex, April 2005. IEEE Computer Society.
9. P. Perla. *The Art of Wargaming*. Naval Institute Press, Annapolis, Maryland, USA, 1990.
10. M. Sakuta. *Deterministic Solving of Problems with Uncertainty*. PhD thesis, Shizuoka University, Japan, 2001.
11. C. Shannon. Programming a computer for playing Chess. *Philosophical Magazine (Series 7)*, pages 256–275, 1950.
12. K. Thompson. Computer Chess Strenght. In M. Clarke, editor, *Advances in Computer Chess 3*, pages 55–56. Pergamon, 1982.
13. E. Zermelo. On an Application of Set Theory to the Theory of the Game of Chess. In *Proceedings of the Fifth International Congress of Mathematicians*, pages 501–504, Cambridge, UK, 1913.