Available online at www.sciencedirect.com

SCIENCE DIRECT®

Theoretical Computer Science 349 (2005) 230–244

Theoretical Computer Science

www.elsevier.com/locate/tcs

# An evaluation function for the game of amazons

Jens Lieberum

*Buckmatten 2c, D-79639 Grenzach, Germany*

**Abstract**

Amazons is a fascinating game that shares properties of chess and Go. Designing a computer program that plays amazons on the level of human experts and beyond is a real challenge. This article emphasizes the secret of such a program, viz. its evaluation function. We describe the function by using explicit formulas, we mention the ideas and goals behind these formulas, we discuss possible refinements, and study in detail methods for special endgame problems. By analyzing a tournament game of AMAZONG against the threefold computer world champion 8QP we illustrate how the new features of our evaluation function can lead to victory.


*Keywords:* Amazons

## 1. Introduction

Amazons is a many-faceted game. The game set typically used to play Amazons is a draughts board of size $10 \times 10$, four white and four black chess queens (called amazons), and a supply of go pieces of one colour (called arrows). The starting position and a first move of white are shown in Fig. 1. A move consists of two steps: (1) the player chooses an amazon of his colour and moves it like a chess queen diagonally, vertically, or horizontally as far as he prefers, provided that no obstacle (another amazon or an arrow) blocks the way; (2) the amazon played has to throw an arrow. Arrows also move like chess queens. They stay at their destination square for the rest of the game and are represented by black squares in the other figures of this article. The players move alternately until one player can no longer move. This happens after at most 92 moves. The player who makes the last move wins the game. A challenging problem concerning the rules of amazons is as follows: Should white's advantage of making the first move be compensated by a komi and if so, by how many points? We will discuss two possible komi rules in more detail in Section 8 of this article.

I first heard about amazons at a workshop on combinatorial game theory at MSRI in July 2000. I was fascinated by the deepness and subtlety of 'simple' positions in amazons that have been analysed by Berlekamp [1], Snatzke [12,13], Müller and Tegos [11]. Inspired by discussions with Müller about his computer program ARROW and my experiences of playing amazons I started to write the computer program AMAZONG. AMAZONG has won the amazons tournaments at the seventh Computer Olympiad in Maastricht in 2002 and at the eighth Computer Olympiad in Graz in 2003. The reader is invited to play against the Java applet AMAZONG at http://jenslieberum.de/amazong/amazong.html.

*E-mail address:* jenslieberum@yahoo.com.

[1] For brevity we only use the male pronoun, where male and female forms are equally applicable, such as in player. Please note, an amazon is a 'she'.

0304-3975/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2005.09.048

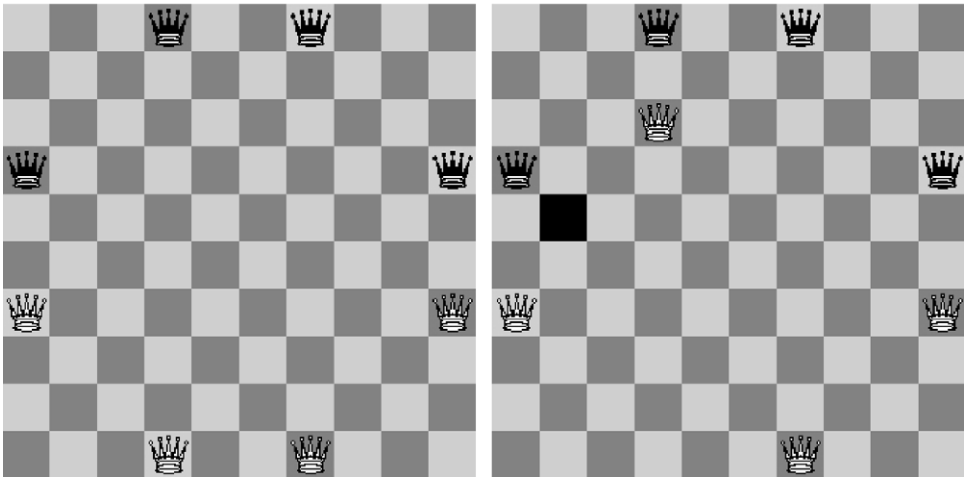

Fig. 1. One good first move out of 2176 possible ones.

I described the general design of my program with a special focus on selective search in talks at the Universities of Jena and Edmonton [6]. This article complements these talks and concentrates on AMAZONG's evaluation function that causes its characteristic style of play. This style clearly distinguishes it from other programs, and is probably its main strength.

Like in many games, the game of amazons can be subdivided into different phases; they are opening, middle game, and endgame. The opening in amazons is a grand challenge for computer programs owing to the absence of opening theory, a branching factor of more than 1000, many positions with more than 20 reasonable moves, and the need for calculating deep variations. Human play is still superior to computers in the opening. At the Computer Olympiad in Graz, AMAZONG used a machine-generated opening book. Our algorithm for building this book combines ideas of Lincke [8] with conspiracy numbers. However, the benefit of opening books is limited in amazons because of the huge complexity of this game. Our evaluation function contains parts that are particularly useful in the opening. We will not treat the automatic generation of opening books in this article.

It is difficult to say exactly when the endgame begins in amazons, but this phase should at least include the so-called *filling phase*. The filling phase consists of those positions where each empty square on the board can be reached by at most one player by some sequence of moves. In most games this happens after approximately 50 moves. We will include problems that appear most frequently in or shortly before the filling phase in our investigation of the endgame.

The article is organized as follows. Sections 2–5 are dealing with the description of different parts of our evaluation function, viz. parts that are most important in the opening and in the middle game. In Section 2 we describe the main ingredients of the evaluation function that are based on king distances and queen distances and combine them into a measure $t$ for positional and territorial evaluation. Section 3 describes why $t$ often does not evaluate positions correctly where the mobility of a single amazon is low. We introduce a new ingredient $m$ for our evaluation function that deals with this problem. In Section 4 we analyse the behaviour of the separate parts of our evaluation function $t + m$ during a game against the computer program 8QP. We find some evidence for the assumption that 8QP lost because it did not consider king distances and mobility. Section 5 contains a collection of ideas for further improvement of the evaluation function.

Sections 6–9 are dedicated to the treatment of special endgame problems. Often, the outcome of the game is clear when the filling phase starts. However, sometimes the positions can involve difficult filling puzzles. They are treated theoretically and on a heuristic level in Section 6. A good solution of the filling puzzle is the basis of a successful treatment of more complex endgame problems. In Section 7 we address a problem that typically appears earlier in the endgame: we explain why amazons that guard a separate region of squares require special treatment in the context of our evaluation function $t + m$. Sections 8 and 9 are dealing with zugzwang positions that would require impracticably deep minimax search to be evaluated correctly by $t + m$. In Section 10 we conclude by reviewing briefly the major problems and challenges of designing an evaluation function for the game of amazons.

## 2. Territorial and positional evaluation

The goal of the game amazons is to have access to more empty squares in the filling phase than the other player. When player $j$ ($j \in \{1, 2\}$, player 1 is white) has exclusive access to a region of $n$ squares, we count these squares as $n$ secure points of the territory of player $j$. When both players can reach a square by some sequence of moves, it is more complicated to guess which player will eventually shoot at that square. For this purpose AMAZONG uses heuristics based on the following ways to measure distances on an amazons board.

Define the distance $d_1(a, b)$ of two squares $a$ and $b$ as the minimal number of chess queen moves needed to go from $a$ to $b$. When there is no path, let $d_1(a, b) = \infty$. Similarly, define the distance $d_2(a, b)$ as the minimal number of chess king moves needed to go from $a$ to $b$. Obviously, we have $d_1(a, b) \leqslant d_2(a, b)$. The distances of player $j$ from square $a$ are then given by

$$D_i^j(a) = \min\{d_i(a, b) \mid \text{the square } b \text{ is occupied by an amazon of player } j\}.$$

On the left (respectively, right) side of Fig. 2 you find an example of $D_1^j(a)$ (respectively, $D_2^j(a)$). In Fig. 2 the upper left corners of empty squares contain the values $D_i^1(a)$ and the lower right corners contain the values $D_i^2(a)$.

All amazons programs seem to use $D_1^j$ in one or another way (for example, see [5]). The idea behind the definition of $D_1^j$ is that $D_1^1(a) < D_1^2(a)$ indicates that player one has better access to the square $a$ than player two. One heuristic for estimating the territory of player 1 is to assume that he will eventually shoot to *all* squares $a$ with $D_1^1(a) < D_1^2(a)$. This heuristic works very well shortly before and in the filling phase. A problem of $D_1^j$ at the beginning of the game is that a single amazon of player $j$ in the centre can cause low values of $D_1^j$ on the whole board, but player $j$ cannot move the amazon into all directions at once. Here $D_2^j$ comes in. One advantage of $D_2^j$ is its locality: often amazons have to fulfill a certain task at their position like guarding the territory in their neighbourhood. Then a large value of $D_2^j(a)$ indicates that player $j$ cannot move towards the square $a$ without causing positional damage, despite a possibly low value of $D_1^j(a)$. Another advantage of $D_2^j$ over $D_1^j$ is that it is more stable when the other player moves and shoots, especially when there are just a few arrows. This makes $D_2^j$ useful for long term estimates and will stabilize the evaluation function in the beginning of the game.

We use $D_i^j$ to assign local evaluations between $-1$ and 1 to each empty square. Positive values indicate an advantage of player 1. Then we sum these numbers over all empty squares in order to transform the local evaluations into global ones. One possible formula for global evaluations $t_1, t_2$ is given by

$$t_i = \sum_{\text{empty squares } a} \Delta(D_i^1(a), D_i^2(a)),$$

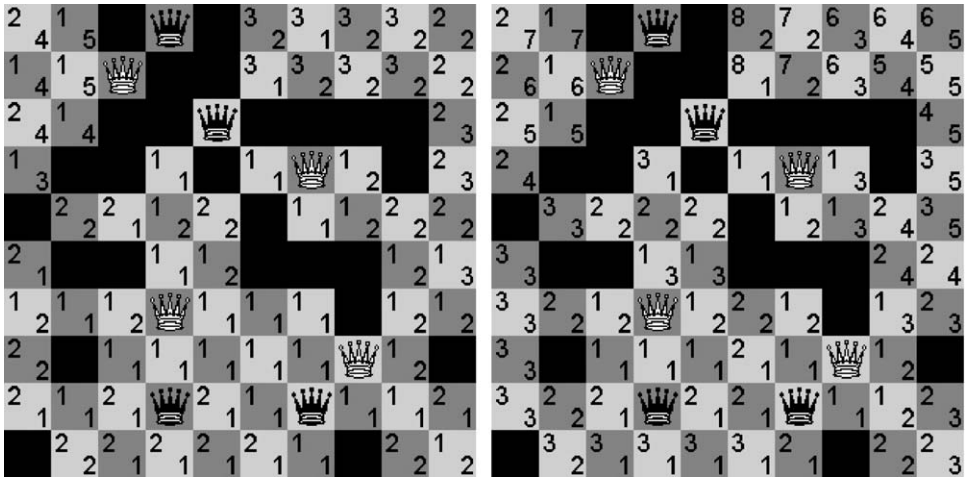

Fig. 2. The minimal distances $D_i^j(a)$.

where

$$\Delta(n,m)=\begin{cases}0 & \text{if } n=m=\infty,\\ \kappa & \text{if } n=m<\infty,\\ 1 & \text{if } n<m,\\ -1 & \text{if } n>m,\end{cases}$$

and $-1<\kappa<1$ is a constant with $(-1)^j\kappa\leqslant 0$ when it is player $j$'s turn. The number $|\kappa|$ estimates the advantage of moving first when the distances of both players to an accessible square agree. We had good experience with $|\kappa|\leqslant\frac{1}{5}$, but some fine-tuning is necessary after each modification of the evaluation function. We optimized the choice of $\kappa$ in order to obtain a low volatility of the evaluations during iterative deepening. This should help to avoid odd–even effects and supports aspiration search with narrow $\alpha$–$\beta$ windows (see [10]).

A program that uses the territorial evaluation $t_1$ as its evaluation function already plays quite reasonably, especially shortly before the filling phase. In contrast to that we observed that the value $t_2$ is useful in the beginning of the game but becomes less significant as the game goes on. The evaluations $t_i$ do not take into account that large positive values of $D_i^2(a)-D_i^1(a)$ indicate a larger advantage for player 1 than small positive values only. This drawback of $t_1$ and $t_2$ is more severe at the beginning than near the end of an amazons game. Therefore, besides $\Delta$, other local evaluations seem to be important as well. The generic approach to find more good local evaluations is to use some array of parameters instead of $\Delta(n,m)$ and then to optimize these parameters. We had good experience with the choices

$$c_1=2\sum_{\text{empty squares } a}2^{-D_1^1(a)}-2^{-D_1^2(a)},$$

$$c_2=\sum_{\text{empty squares } a}\min(1,\max(-1,(D_2^2(a)-D_2^1(a))/6)).$$

Notice that in $c_1$ the local advantage $(D_1^1(a),D_1^2(a))=(1,2)$ is rewarded by 0.5 points for player 1, $(2,3)$ by 0.25 points, $(1,3)$ by 0.75 points, and squares $a$ with $(D_1^1(a),D_1^2(a))=(n,n)$ contribute 0 points. Other tuples are of minor practical importance for $c_1$. In contrast to $c_1$, $c_2$ depends only on $D_2^2(a)-D_2^1(a)$ and only large differences indicate a clear advantage of one player.

Now we have to combine the values $t_i$ and $c_i$ into one evaluation function. A weighted sum with static weights does not seem to be appropriate for this because the importance of the values $t_i$ and $c_i$ varies during the game. Therefore, we define

$$w=\sum_a 2^{-|D_1^1(a)-D_1^2(a)|},$$

where we sum over all empty squares $a$ with $D_1^1(a)<\infty$ and $D_1^2(a)<\infty$. Obviously, we have $w=0$ if and only if the position belongs to the filling phase and typically $w$ decreases with the number of moves played. These properties of $w$ are useful for our purposes. In particular, when the game approaches the filling phase it will be much better to use $w$ instead of other parameters such as the number of arrows on the board. Of course, the definition of $w$ is by no means canonic and leaves ample room for optimizations and new ideas. Continuing along these lines of thought we define an evaluation $t$ as

$$t=f_1(w)t_1+f_2(w)c_1+f_3(w)c_2+f_4(w)t_2,$$

where $(f_i)_i$ is a partition of 1 (meaning $0\leqslant f_i(w)$ and $\sum_i f_i(w)=1$). The exact form of the functions $f_i$ is a problem of parameter optimization. Our choice of $f_1$ has been guided by the observation that $t_1$ becomes increasingly important during the game and gives quite good estimates of the expected territory shortly before the filling phase. Therefore, $f_1$ is monotonously decreasing and satisfies $f_1(0)=1$. The counterpart of $t_1$ is $t_2$. It rewards balanced distributions of the amazons of one player and helps to hinder the other player from reaching such a distribution. This is most important at the beginning of the game. The values $c_1$ and $c_2$ allow detection of finer properties of the position than $t_1$ and $t_2$ alone, because they depend on the quality of local advantages. They support good positional play in the opening and a smooth transition between the beginning and later phases of the game. This is most evident for $t_1$ and $c_1$: while at the end of the game only $t_1$ counts, $c_1$ rewards moves in earlier phases of the game that replace clear local disadvantages by small disadvantages and small advantages by clear advantages.

## 3. Mobility of individual amazons

AMAZONG tries to enclose amazons of the other player inside of small regions at the beginning of the game. Compared to other computer programs, this is AMAZONG's main strength. In this section we will present a modification of the evaluation function $t$ (see Section 2) that is responsible for this behaviour.

Enclosing amazons typically does not cause an appropriate change of $t$ (and especially of $t_1$) in the beginning of the game. This can be explained as follows: when a single amazon $A$ of player 1 is enclosed in some small region of $n$ points, then the amazons board is divided into two parts: the inside and the outside of that region. Player 1 has exclusive access to the territory on the inside. This contributes $n$ points to $t$. On the outside, some active amazons of player 1 might overshadow the missing influence of $A$ in $D_1^1$. In addition, some amazons of player 2 that have helped to enclose $A$ might not be in optimal positions but often have a large potential to improve their positions. The problem that $A$ cannot reach the outside for the rest of the game is not reflected in the computation of $t$. The disadvantage of the enclosed amazon often starts to affect $t$ several moves later. Then it is too late. Therefore, a correction term $m$ is needed to take into account the mobility of individual amazons. Since active amazons can overshadow bad positions of passive amazons in the evaluation function $t$ it seems more important to punish passive and enclosed amazons than to support active amazons in this correction term. To compute $m$ quickly, consider first the number $N(a)$ of empty squares that can be reached from $a$ by a single move of a chess King. The numbers $N(a)$ can be updated incrementally during the search inside of functions doMove and undoMove. For an amazon $A$ of player $j$ on the square $a$, let

$$\alpha_A = \sum_b 2^{-d_2(a,b)} N(b),$$

where we sum over all squares $b$ with $d_1(a,b) \leqslant 1$ and $D_1^{3-j}(b) < \infty$. When $\alpha_A = 0$ we say that the amazon $A$ is enclosed. Examples of the values $N(a)$, $\alpha_A$, and of enclosed amazons are shown in Fig. 3. For example, for the white amazon $A$ in the upper left corner of Fig. 3 (left), we compute $\alpha_A = 7+6+5+3+3+(5+4+7+4)/2+5/4 = 35.25$. The two white amazons in the lower right corner of this figure are enclosed.

We have learned in discussions with experienced amazons players that at the beginning of a game on a board of size $10 \times 10$ enclosed amazons should be punished by a malus of at least 10 points. In general, we use $w$ from the last section to define

$$m = \sum_{\substack{\text{amazons } B \\ \text{of player 2}}} f(w, \alpha_B) - \sum_{\substack{\text{amazons } A \\ \text{of player 1}}} f(w, \alpha_A)$$

for a suitable function $f \geqslant 0$. The exact choice of $f$ is the hardest optimization problem in our evaluation function $t+m$, so we restrict our description to the properties of $f$ that did not change during our experiments: $f$ satisfies $f(0, y) = 0$

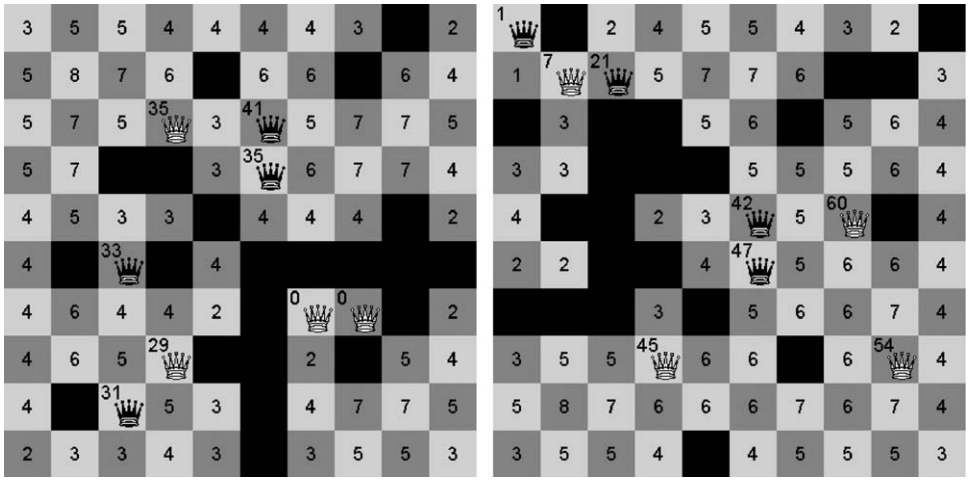


Fig. 3. Neighbours $N(a)$ of empty squares $a$ and the values $\alpha_A$.

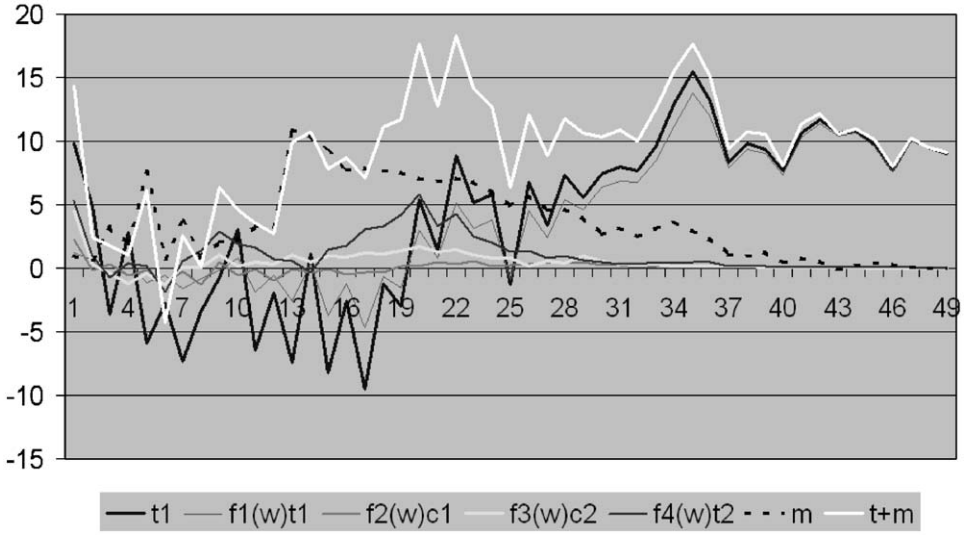


Fig. 4. The components of the evaluation function $t + m$ during a game.

.

and $(\partial f/\partial x)(x, y) \geqslant 0$ because the longer an amazon is enclosed before the filling phase starts the larger is the disadvantage. Furthermore, $f$ satisfies $(\partial f/\partial y)(x, y) \leqslant 0$ because a low value of $\alpha_A$ corresponds to a passive position of amazon $A$. The last dependence is not linear. We had commendable experience with functions $f$ that satisfy $2f(w, 5) < f(w, 0)$. This can be explained as follows: $\alpha_A \approx 5$ indicates that amazon $A$ is almost enclosed. However, there is a large difference between an enclosed and an almost enclosed amazon. The other player possibly has to move one of his own amazons to an unfavourable square to prevent $A$ from escaping. The resulting change of $t$ then has to be compensated for by $m$. In addition, the task of guarding not completely enclosed amazons makes the guarding amazons less mobile and more vulnerable for attacks.

The big difference between enclosed and almost enclosed amazons can be seen on the right side of Fig. 3. White can enclose the black amazon $B$ with $\alpha_B = 1$ in his next move, but then Black can reply by enclosing the white amazon, too. Similarly, the task of guarding the white amazon in the upper left corner puts amazon $B$ with $\alpha_B = 21$ in danger of getting enclosed.

## 4. Comparison between $t_1$ and $t + m$

In this section we compare our evaluation function $t + m$ with $t_1$ by using the game AMAZONG vs. 8QP played at the 7th Computer Olympiad in Maastricht. The position after 26 moves in this game is shown in Fig. 2. AMAZONG won the game by 8 points, mainly due to the enclosed black amazon in the upper left corner. Fig. 4 shows how $t_1$ and the different components of $t + m$ varied during the game. The values $t_i$ are computed using $|\kappa| = 0.1$. The lines corresponding to $t_1$, $m$ and $t + m$ are clearly visible. On move 13 white enclosed the black amazon which causes the maximum of the dashed line corresponding to $m$. Notice that at this point the evaluation $t + m$ predicts the outcome of the game very well and differs from $t_1$ by more than 18 points. After move 13 $t_1$ and $t + m$ become increasingly related and finally coincide when the filling phase is reached.

As expected, the values $c_2$ and $t_2$ are more stable than $c_1$ and $t_1$. In addition, $c_2$ and $t_2$ are positive in almost all positions of the game. This indicates that the evaluation function of 8QP does not consider king move distances. Therefore, 8QP puts up no resistance against AMAZONG maximizing these components of $t + m$.

## 5. Refinements

Below we consider positions with regions that are (almost) separated by arrows. An intriguing question is: how much is it worth when one player has a majority of amazons inside such a region? Instead of looking for a general

answer to this difficult question, we simply observe that the territorial evaluation $t$ has the tendency to underestimate the advantage of the majority. Below we deal with five possible refinements or improvements. The first idea is that a possible correction term of $t$ could take into account the distances between each empty square and each amazon. However, the computation of these values would take almost four times longer than the computation of $D_i^j(a)$. Therefore, it seems more appropriate to compute only the numbers of amazons $A_v$ of player $j$ on squares $b_v$ that satisfy $d_i(a, b_v) = D_i^j(a)$. These numbers can be computed efficiently together with $D_i^j(a)$. They are useful as additional inputs for refined definitions of $c_i$ and $t_i$. In addition to these corrections, the disadvantage of having a majority of amazons in a *small* region early in the game should be reflected by $m$. This situation is not treated correctly by $m$ because when amazons of both players are inside one region the involved amazons are not considered as being enclosed.

A second refinement is as follows. In some experiments, we weighted squares in the computation of $c_i$ and $t_i$. The weights depended on $w$ and the distance of the square from the centre of the board. It is difficult to assess the importance of this second refinement.

A third idea for improvements is to repeat the constructions of Section 2 for other distance functions such as $d_1 + d_2$ or $2d_1 + d_2$ (or estimates of these distances that can be computed more efficiently). One has to decide very carefully how many different distance functions one should use, because each additional distance function slows down the evaluations considerably.

A fourth refinement concerns the distribution of amazons on the board. In the opening it is desirable (especially for black) to reach a position with exactly one amazon in each corner of the board. The distances from such a distribution can be used to improve the evaluation function in the opening phase (see [7]).

Partially, the fourth refinement helped to fix the biggest weakness of our evaluation function, namely the underestimation of large territorial frameworks at the beginning of the game. In a fifth refinement, we deal with the same problem by adding a bonus for huge areas of potential territory (see [9]).

## 6. The filling puzzle

Some positions in amazons endgames require a very deep search. Other endgame positions can be evaluated correctly by humans without deep search but would require a 10-ply or even a 40-ply deep search by a program with the evaluation function described so far. We will address the latter endgame problems in this article.

The simplest endgame positions arise when the board is completely decomposed, meaning that amazons of different colours are separated by arrows. The game then consists of two puzzles: filling of white and black territory. Buro [3] showed that the filling puzzle is NP-hard. In contrast, most positions that arise in actual games on a board of size $10 \times 10$ can be filled completely. Below we will describe a simple heuristic that often succeeds in filling the maximal possible number of squares. For the cases where our heuristic fails or where complete filling is not possible, more sophisticated methods are required. For these cases we will propose some ideas for a better heuristic solution of the filling puzzle. Such a solution would be helpful in Sections 7 and 9. Moreover, it would help to find the right moment to stop playing in a game of amazons without relying on human help and continuing with wrong assumptions. This would make amazons programs and amazons game servers more user friendly.

Typically, we assume that a connected region of $n$ empty squares containing amazons of one player $p$ provides a resource of $n$ moves for the player $p$. If the connected region only provides a resource of $m < n$ moves, then we call this region *defective territory* and call the number $n - m$ the *defect* of the region. A simple way that helps to avoid the creation of defective territory is to add to the usual evaluation function a term $s$ that measures the shape of the territory. Starting to avoid the creation of defects in the filling phase might be too late. Therefore, one starts to compute $s$ when $w$ (see Section 2) drops below some threshold value. In order to avoid interference of $s$ with more important ingredients of our evaluation function, we choose $s$ to be very small. The definition of $s$ is based on two heuristics. First, we assume that it is typically simpler to fill a region with more than one amazon than with a single amazon. This observation leads us to the following definitions: we define $M_1$ (resp. $M_2$) as the set of all squares $a$ such that the queen distance of player one from $a$ is smaller (respectively, larger) than the queen distance of player two from $a$, or the square $a$ is occupied by a white (respectively, black) amazon. Second, let $s_1$ be the number of connected components of $M_2$ minus the number of connected components of $M_1$, where we consider squares of king distance one as connected. Using the

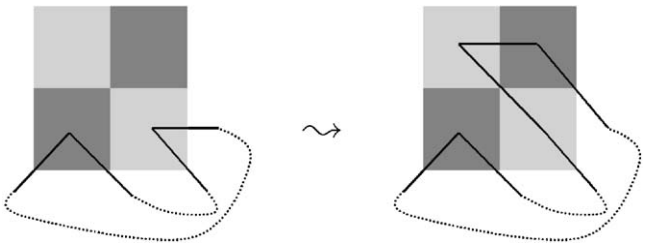

Fig. 5. Extending the path to a new 2 × 2-block.

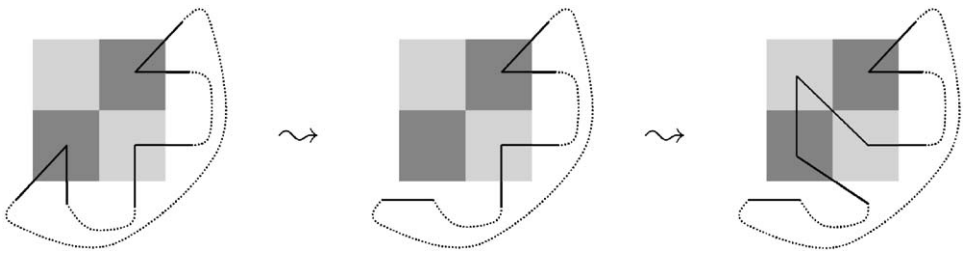

Fig. 6. Modifying the old path on the outside first.

number $N(a)$ of neighbours of a square $a$ from Section 3 we define

$$s = s_1/1000 + s_2/10\,000 \quad \text{where } s_2 = \sum_{a \in M_1} N(a) - \sum_{b \in M_2} N(b).$$

The heuristic value $s = s_2/10\,000$ alone also leads to good results in many cases. The idea behind the definition of $s_2$ is that high local connectivity helps to avoid defects. The following theorem gives some justification of this second heuristic behind the definition of $s$.

**Theorem 1.** *If a region R can be built up step by step by starting with a single* $2 \times 2$*-block and by adding* $2 \times 2$*-blocks that overlap with the preceding region, then R can completely be filled by a single amazon starting on some square of R.*

**Sketch of Proof.** We will prove the stronger statement that the amazon can traverse each square of the region once and then end up on her starting square by making king moves. This will imply that for any starting square the amazon can follow a circular path and always shoot to the square where she came from, thereby filling the whole region. This will prove the theorem. In order to simplify the proof of the stronger statement we also require that the circular path can be drawn on the board without self-intersections.

The stronger statement is now proved by induction on the number $k$ of $2 \times 2$-blocks. The case $k = 1$ is trivial. We illustrate the induction step by examples: the left side of Fig. 5 shows a path that goes through two squares of a new $2 \times 2$-block. The directions of the lines in the picture inform us about the neighbouring squares on the outside of the new block that the path traverses before entering and after leaving the block. The right side of Fig. 5 shows how the path can be extended to the whole block without modifying the path on the outside.

In some cases the old path cannot simply be extended to the new $2 \times 2$-block. In these cases, the old path also has to be modified on the outside of the new block. An example is shown in Fig. 6: on the left side of Fig. 6, the squares where the path enters and leaves on the lower left side are neighbours. Therefore, it is possible to find a shortcut on the outside as shown in the middle of Fig. 6[2]. Then the path can be extended to the new block as before.

It is an easy (but long) exercise to extend Figs. 5 and 6 to a complete list of cases. □

[2] Here our assumption that the path has no self-intersections is important to ensure that this is possible in all remaining cases.

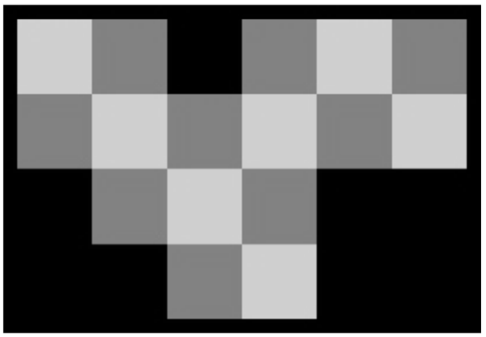

Fig. 7. By Theorem 1 this region has no defect.

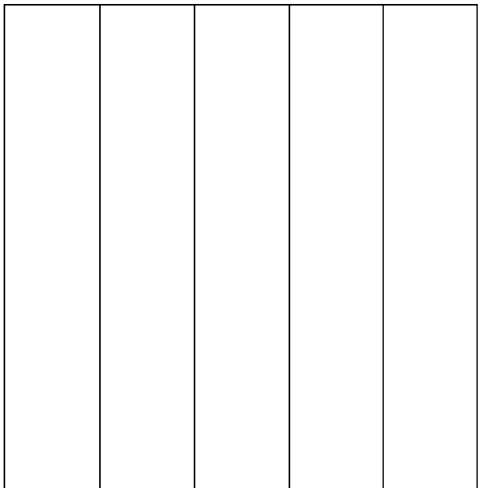

Fig. 8. A blueprint of a region with a large defect.

An example of a region that can be filled completely according to Theorem 1 is shown in Fig. 7.

Regions with large defects are rare, but they exist. The following theorem implies the existence of regions with a defect that is larger than 99.999% of the size of the region.

**Theorem 2.** *Define numbers $k_n$ such that the largest defect for connected regions with n squares and one amazon is equal to $n - k_n\sqrt{n}$. Then there exist $a, b \geqslant \frac{1}{2}$ such that $k_n \in [a, b]$ for all n.*

**Proof.** For a region containing $k \times k$ vertices in the planar grid we construct a trivalent graph $G_k$ that has the shape of a comb as shown in Fig. 8 and that goes through all vertices of the region. The longest path $L_k$ inside $G_k$ has length $3k - 3$.

Then we transform the graph $G_k$ into a region on an amazons board by following a procedure of Buro[3]: in the planar grid we replace each vertex and its adjacent half-edges by a certain region of squares on an amazons board of size $9 \times 9$. For the 9 vertices of the graph $G_3$ we show the corresponding regions in Fig. 9 (compare Figs. 9 and 10 of [3]). By sticking together the regions of size $9 \times 9$, $G_k$ is transformed into a region of squares $R_k$ on an amazons board of size $9k \times 9k$ consisting of

$$r_k = 5k + 10(k-2)k + 14(k-2) + 16 = 10k^2 - k + 12$$

squares ($k > 1$). We define $R_1$ to be a single square.

[3] More precisely, our description is equivalent to Buro's transformation with corridor length 2.

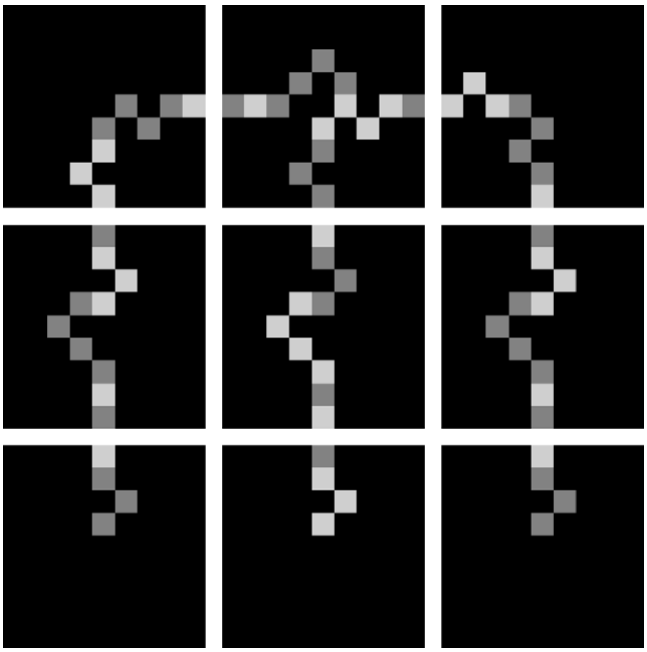

Fig. 9. Replacing the vertices of $G_3$.

The regions from Fig. 9 have the property that an amazon that traverses a region has to block a distinguished central square with the consequence that each region can be traversed at most once. Therefore, it is easy to see in our case that the image $F_k$ of the longest path $L_k$ under Buro's transformation is the maximal region of squares of $R_k$ that can be filled by a single amazon inside of $R_k$.[4] For $k > 1$ the path $F_k$ consists of $f_k = 10 + 30(k-2) + 16 = 30k - 34$ squares.

For all $n$ with $r_k \leqslant n < r_{k+1}$ we simply add squares to $R_k$ (say at the lower right corner) in a way that we obtain regions $Q_n \supset R_k$ of size $n$ such that $(Q_n \setminus R_k) \cup F_k$ is the largest region of $Q_n$ that can be filled by a single amazon. This largest region contains at most $p_n = f_k + r_{k+1} - r_k - 1$ squares. We have $p_n = \Theta(\sqrt{n})$ so, in particular, $p_n < b\sqrt{n}$ for some constant $b$ and for all $n \geqslant 1$. Since $n - b\sqrt{n}$ is a lower bound for the largest possible defect, the number $b$ is an upper bound for all $k_n$ defined in the theorem.

In addition, for every square of a connected region of size $n$ on an amazons board we can find a path that starts at the given square and contains at least $\frac{1}{2}\sqrt{n}$ squares. This implies the existence of the lower bound $a$ for the sequence $(k_n)_n$. □

The smallest regions with defects are shown in Fig. 10. In all three cases, white has to throw an arrow to the square in the center in his next move, so white can only fill one square.

A simple solution of the filling problem uses $s$ from above together with some knowledge about small defective territories. This solution is easy to implement and of high practical value. For an implementation of an endgame database including more knowledge about defective territories than shown in Fig. 10, a data structure called *line segment graph* is useful (see [11]).

For larger regions endgame databases can only contribute shortcuts near the end of the search and local properties are not sufficient to obtain good heuristics for filling. For example, consider a connected region $R = R_1 \cup R_2$ with disjoint connected parts $R_1$ and $R_2$. If region $R_2$ has the property that once we enter $R_2$ we can no longer come back to $R_1$, it makes sense to fill as much of $R_1$ as possible before going to $R_2$. But if we can come back from $R_2$ to $R_1$ a completely different plan might be appropriate. Therefore, we believe that besides local connectivity heuristics a global plan that guides the heuristic search is important. A 'global plan' might mean to specify for each square how urgent

[4] Buro used long corridors to enforce this property for arbitrary trivalent graphs in the planar grid. This is not necessary in our case.

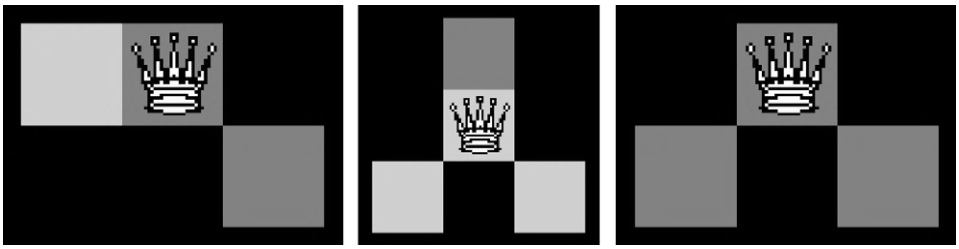

Fig. 10. Small defective territories.

it is to fill it during heuristic search. One way to assign these urgency values uses the results of the search so far: one first determines the 'difficulty of a square' depending on the number of times the square under consideration could be filled during preceding searches and then determines (for example) randomly if certain regions of difficult squares should be filled with high or with low urgency. In this context, Monte-Carlo simulations seem to be a good choice for the heuristic search, because the expense of this method can nicely be scaled. This is helpful for trying out different global plans and for implementing a smooth transition between minimax search and single agent search.

The method above finds a lower bound for the number of squares that can be filled. Often this lower bound is equal to the size of the region and we are done. General methods to find good upper bounds for the filling puzzle would be very useful. The investigation of so-called dead ends in [11] has been a first step in this direction.

## 7. Guards

In this section we propose an improvement of our evaluation function in positions where amazons guard certain pieces of territory. Such amazons can already appear very early in a game, but it is almost certain that they will appear in the endgame some moves before the filling phase starts.

Consider amazon $A$ on the square c6 in Fig. 11. She has exclusive access to 10 points of territory in the upper left corner. In addition, she has direct *outside influence* on the squares c5, c4, c3, c2, c1, d5, e4, d6, e6, f6, g6, h6, and i6. The problem with the position in Fig. 11 is that amazon $A$ has to give up the upper left corner if she wants to make use of her outside influence by moving to one of these squares. Therefore, the part $t_1$ of our evaluation function (see Section 2) tends to overestimate the outside influence of amazon $A$. In addition, it would not be appropriate if one would completely neglect the outside influence of $A$. This influence restricts the mobility of the white amazons on d7 and d3 in Fig. 11. The amazon on d7 threatens to move to c6 when amazon $A$ leaves. The amazon on d3 has to pay attention to the threat that $A$ moves to the lower left corner and shoots to c6.

The preceding observations lead us to the following definition: we say that an amazon is a *guard* (or *n-guard*), if she has exclusive access to $n > 0$ squares of territory. For an $n$-guard $A$ of colour $c$ we count the number $n_A$ of empty squares $a$ such that $A$ is the only amazon of colour $c$ in queen distance one of $a$ and the opponent is in queen distance one of $a$, too. We define $g_A = \min(n, n_A/2)$ and correct our evaluation function by subtracting $g_A$ for all white guards $A$ and by adding $g_A$ for all black guards $A$. For example, in Fig. 11 we have $g_A = 3.5$ for the guard $A$ on c6. In this example, the seven squares c2, c3, c4, d5, e4, d6, e6 contribute to $n_A$. We note that the square i6 does not contribute to $n_A$ because the black amazon on h7 is also in queen distance one of i6.

We do not require that a guard cannot enlarge its territory by some move. Partially, our evaluation of guards takes care of this problem. In addition, problems of this kind are easily resolved by search. Also notice that two amazons of player $p$ that have access to the same secure territory are not guards in our sense because we require *exclusive* access of guards to some territory. We insist on exclusive access because a position where player $p$ can choose which amazon should leave a territory typically does not cause a big loss of outside influence at all.

The next section will be concerned with a treatment of zugzwang positions. The concept of the outside influence of guards and a good solution of the filling puzzle are important in a heuristic treatment of these positions. The evaluation of the outside influence of guards that are involved in zugzwang positions will slightly differ from this section.

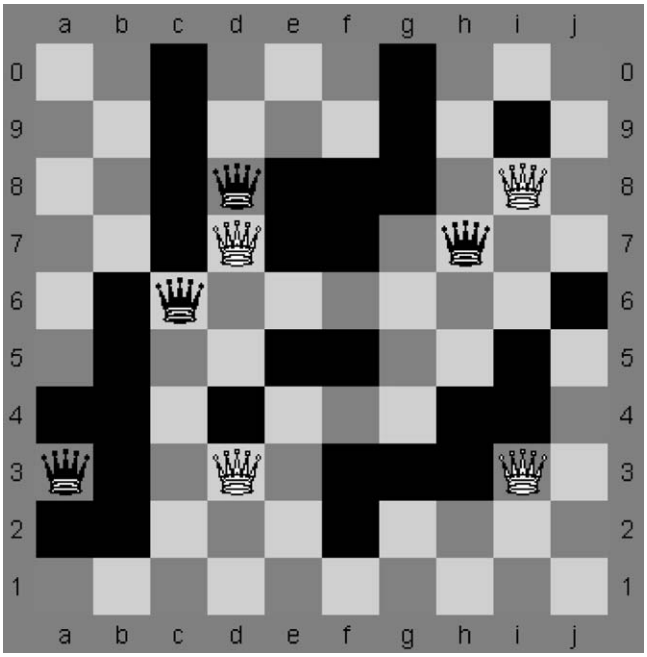


Fig. 11. A guard on c6.

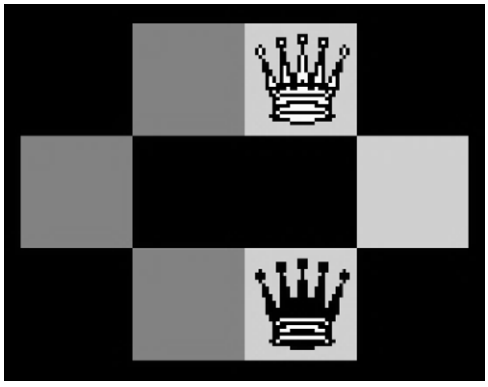

Fig. 12. Black to move, komi 2, who wins?

## 8. Komi and zugzwang

Below we discuss the following interesting question. Who wins in the position shown in Fig. 12 when it is black's turn and the game is played with a komi of two points?

The answer to that question depends on subtle details of the komi rule. Let us consider two versions of giving $n$ komi points to black.

1. Black may pass up to $n$ times instead of making a move.
2. The game is played until one player cannot move. White wins in the case where black cannot move and white can make at least $n$ more moves. Otherwise black wins.

Under most circumstances, rules 1 and 2 lead to the same outcome of a game. Positions involving zugzwang are an exception. Using rule 1 in Fig. 12, black passes and wins. Using rule 2, black has to make the first move and loses.

An implementation of komi rule 2 requires a solution of the filling puzzle. Rule 1 is simpler, looks more natural, and is easy to implement. An implementation of komi rule 1 does not require a modification of the evaluation function. Since a komi of $n$ points in the sense of rule 1 is equivalent to an additional black amazon inside of an isolated piece of territory of $n$ points, we can think of a komi in this sense as a part of the position.

## 9. A generic zugzwang position

The position in Fig. 12 will be evaluated as a small advantage for the player to move by our evaluation function. This is wrong, but it is only slightly wrong. We are looking for zugzwang positions where our actual evaluation function goes wrong completely because in these cases a larger improvement of our evaluation function is possible. In addition, we want to restrict attention to positions that often show up in games and where a heuristic evaluation does not involve too much search. A representative example of such a position is shown on the left side of Fig. 13. Our evaluation function assigns an advantage of 4 points for white to this position. The reader should verify that the player to move in the position loses. Therefore, 0 would be a better evaluation of the position.

Now we will define *generic zugzwang positions* by using the picture on the right side of Fig. 13. We think of $R_1$ (resp. $R_2$, $R_3$, $R_4$) as regions of empty squares, where regions $R_1$, $R_2$ are adjacent to a white amazon and regions $R_3$, $R_4$ are adjacent to a black amazon. It is not necessary that the two amazons are diagonally adjacent to the regions $R_i$ and some regions may be empty. Denote the square with the white (respectively, black) amazon by $W$ (respectively, $B$). In difference to the picture, we require in our definition only that the squares $W$ and $B$ are in king distance 1. Let $a$ (respectively, $b$) be the number of moves provided by the region $R_1 \cup W \cup R_2$ (resp. $R_3 \cup B \cup R_4$). We may assume that $a \geqslant b$. Then we say that the position is a generic zugzwang position if the following conditions are satisfied:

- the white (respectively, black) amazon has *exclusive* access to the *separate* regions $R_1$ and $R_2$ (respectively, $R_3$ and $R_4$),
- $R_1 \neq \emptyset$ and $R_2 \neq \emptyset$,
- for all best white (respectively, black) moves in the position the arrow is thrown to $W$ (respectively, $B$).

We now consider the examples of zugzwang positions in Fig. 14 by going from the left side to the right side in this figure. The first two positions are generic zugzwang positions. For the third and fourth position, the last condition from our list does not hold true. In both cases, the black amazon can go one step downwards and throw an arrow to the lowest square of the picture. Then the white amazon must not move to the square $B$ because by throwing an arrow she would block her way back. Therefore, the third and fourth position in Fig. 14 are not generic zugzwang positions.

Consider a generic zugzwang position with $a \geqslant b$ as above. Define $c_i$ $(i = 1, 2)$ as the maximal number of moves provided by the region $R_i$ with a single amazon on some square of $R_i$ in queen distance 1 of $W$. Then a good evaluation of the generic zugzwang position is given by

$$v = \max\{c_1 + 1 - b, c_2 + 1 - b, 0\}.$$

For example, we have $v = 2$ for the first position in Fig. 14 and $v = 1$ for the second position in this figure. The evaluation $v$ coincides with the value of the position in combinatorial game theory [4,2]. When it is important to win by a certain margin, or equivalently, when komi rule 2 of the previous section is used, then the position is more complicated and an analysis of sums of generic zugzwang positions is equivalent to the knapsack problem (see [11]).

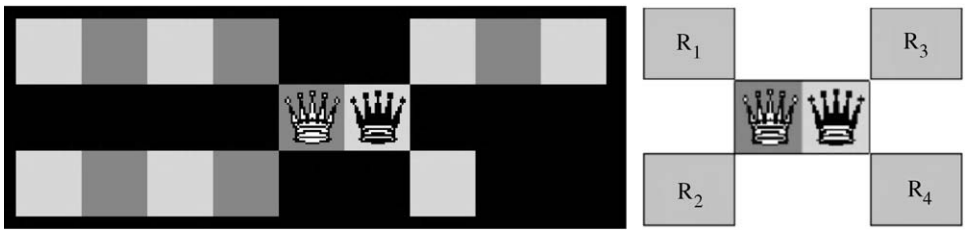


Fig. 13. Abstracting from an example of a zugzwang position.

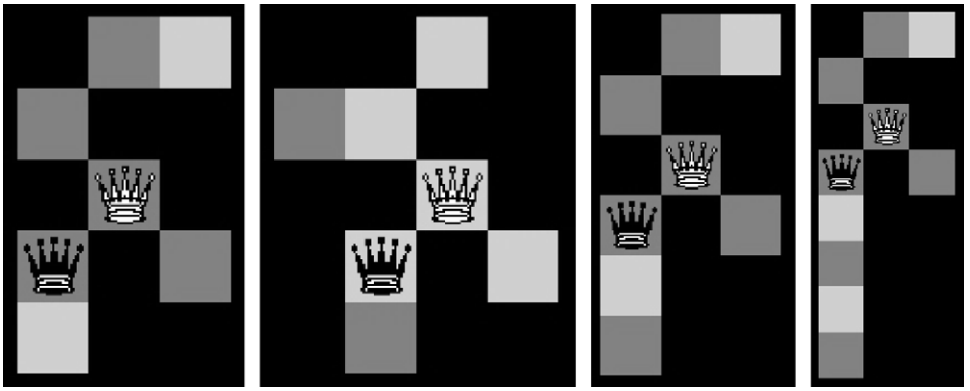

Fig. 14. Which positions are generic zugzwangs?

It is interesting to see how good the evaluation $v$ is in the case of the two positions in Fig. 14 that are not generic zugzwang positions: for the third position in this figure, $v = 1$ gives the correct value of combinatorial game theory, but this position requires a treatment different from the generic zugzwang positions when komi rule 2 is used. In the fourth position, we have $v = 0$, but $-1$ is the correct value.

In practice, we are already satisfied when the evaluation $v$ is typically better than an evaluation that does not consider zugzwang. This allows to loosen some conditions or to replace them by heuristics. For example, we may assume that no defective territory is involved. In other words, let $a_i$ be the number of squares in $R_i$. Then we assume $a = a_1 + a_2$, $b = a_3 + a_4$, $c_1 = a_1 - 1$, and $c_2 = a_2 - 1$. We did not successfully use weaker versions of the conditions for $R_1$ and $R_2$ in the definition of a generic zugzwang position. We had good experience with including positions where amazons have outside influence. For example, if the white amazon has outside influence, then we regard the white amazon as a $\max(a_1, a_2)$-guard and evaluate the outside influence according to Section 7. The black amazon may be inside of some large region together with other black amazons in this example. We treat this case as $R_3 = \emptyset = R_4$, make no adjustments to the evaluation of the influence of the black amazon, but subtract the value $\min(a_1, a_2)$ from our usual evaluation because it is likely that the situation ends up in zugzwang.

## 10. Conclusion

We have described all components of the evaluation function of our program AMAZONG. An important idea in our evaluation function is the smooth transition between many different strategic goals at the beginning of the game and a purely territorial evaluation in the filling phase. The most difficult problems are to find (1) a good balance between mobility and territory and (2) a realistic evaluation of the advantage of local majorities of amazons. In the endgame, a variety of new problems appear that require special treatment. Partially, these problems can be solved by endgame databases. For the remaining positions that involve zugzwang or difficult filling puzzles, global minimax search is not very helpful, but these problems often can be solved by some restricted usage of search inside of the evaluation function.

## References

[1] E.R. Berlekamp, Sums of $2 \times N$ amazons, in: F.T. Bruss, L. le Cam (Eds.), Game Theory, Optimal Stopping, Probability and Statistics: Papers in honor of Thomas S. Ferguson, Institute of Mathematical Statistics, Lecture Notes—Monograph Series, Vol. 35, 2000, pp. 1–34.

[2] E.R. Berlekamp, J. Conway, R. Guy, Winning Ways, Academic Press, London, 1982.

[3] M. Buro, Simple amazons endgames and their connection to Hamilton circuits in cubic subgrid graphs, in: T. Marshland, I. Frank (Eds.). Proc. Second Internat. Conf. Computers and Games, CG00, Vol. 2063, Lecture Notes in Computer Science, 2001, pp. 250–261.

[4] J. Conway, On Numbers and Games, Academic Press, New York, 1976.

[5] T. Hashimoto, Y. Kajihara, N. Sasaki, H. Iida, J. Yoshimura, An evaluation function for amazons, in: H.J. van den Herik, B. Monien (Eds.), Advances in Computer Games, Vol. 9, Universiteit Maastricht, The Netherlands.

[6] J. Lieberum, Selective Search in Amazons, 2002 (powerpoint presentation).
[7] J. Lieberum, An evaluation function for the game of amazons, 2003 (powerpoint presentation).
[8] T.R. Lincke, Strategies for the automatic construction of opening books, in: T. Marsland, I. Frank (Eds.), Proc. Second Internat. Conf. Computers and Games, CG00, Vol. 2063, Lecture Notes in Computer Science, 2001, pp. 74–86.
[9] R. Lorentz, Finding Territory in Amazons, The seventh computer olympiad computer-games workshop proceedings, Technical Reports in Computer Science, Universiteit Maastricht.
[10] T. Marsland, A review of game-tree-pruning, ICCA J. 9 (1) (1986) 3–19.
[11] M. Müller, T. Tegos, Experiments in computer amazons, in: R. Nowakowski (Ed.), More Games of No Chance, 2002, pp. 243–260.
[12] G. Snatzke, Exhaustive search in the game amazons, in: R. Nowakowski (Ed.), More Games of No Chance, 2002, pp. 261–278.
[13] G. Snatzke, New Results in Exhaustive Search in the Game Amazons, University of Jena, 2002, preprint.