

# Enhancing Search Efficiency by Using Move Categorization Based on Various Game Progress Values in Amazons

Yoshinori Higashiuchi

Department of Computer Science, Saga University, Saga, Japan.

Email:hi\_yoshi@fu.is.saga-u.ac.jp

Reijer Grimbergen

Department of Informatics, Yamagata University, Yonezawa, Japan.

Email:grim@yz.yamagata-u.ac.jp

## Abstract

Amazons is a two-player perfect information game with an average number of legal moves that is higher than chess, shogi or Go. Our aim is to improve the search efficiency of an Amazons program by finding good moves to search first, thus improving the efficiency of  $\alpha$ - $\beta$  search. In earlier work, we established that a static move ordering scheme has important problems and that move ordering needs to be changed as the game progresses. The number of moves from the start of the game was used as a simple method to measure game progress and this improved the playing strength of our Amazons program. However, move number might be a too simplistic method and positions where using move number as progress value gives the wrong results can easily be constructed. In this paper we propose three other methods to measure game progress: using territory, using mobility and a combination of territory and mobility. We then compare the performance of the four different methods for measuring game progress using self-play experiments. These experiments indicate that territory is the most promising of the four methods, but the results are not clear enough to warrant a definite conclusion.

**Keywords:** Selective search, move categorization, game progress, Amazons.

## 1 Introduction

Amazons is a two-player perfect information game with very simple rules [8]. From a computational point of view, its main feature is the large number of legal moves, particularly in the early stages of the game (in the initial position there are 2,176 possible moves). Even though the number of legal moves decreases as the game progresses, the average number of moves in an Amazons' position (479) [7] is considerably larger than chess (35), shogi (80) or Go (250) [6]. Because of the large average number of moves, the well-known search techniques that have been so successful in other games cannot be easily applied to Amazons. Therefore, it is necessary to find new efficient search methods, which is why Amazons has attracted some attention recently as a topic of research.

In games like chess and Go, centuries of experience have resulted in heuristics for good moves. These heuristics can be used in a game program to search moves with a high potential pay-off early. In chess for example, moves that capture pieces for free or move pieces near the center of the board are

given priority, while moves that lose material are searched last. By using these heuristics, the efficiency of  $\alpha$ - $\beta$  search can be improved, resulting in a considerable speed-up of the search.

In contrast, Amazons is a relatively new game, so there are no known strategies on how to play the opening and there is no expert feedback available to decide which moves are good in the middle game. Therefore, to improve the efficiency of the search in Amazons, first we need to know what good moves are. We believe that blocking Amazons and escape moves from blocks are important features in Amazons. In earlier work [4], we presented a classification of moves based on the number of times the moving Amazon and the arrow that was shot are blocking opponent Amazons. Based on this classification, we presented a move ordering that significantly improved the strength of our Amazons program TAS.

However, we also showed that move ordering is not enough. In the opening of Amazons, the mobility of the pieces is important, while in the endgame the ability to freely move in territories that are larger than that of the opponent is the most im-

portant strategic feature. Therefore, the strategic features shift as the game progresses and the move ordering should reflect this change. By introducing game progress into the move ordering scheme, we were able to further improve the strength of our Amazons program.

In our earlier experiments, we used move number (counting from the start of the game) for measuring game progress. This is a very simple method, but it is also wrong in many cases. For example, a game of Amazons is finished when the territories of all Amazons are mutually exclusive (after this, counting the number of moves it takes to fill the territories is sufficient to decide the winner). The moment this happens is only loosely related to the number of moves played in the game. Although we believe that move number is a natural progress value in most cases, it needs to be compared to other methods to justify any definite conclusion.

Therefore, we need to compare move number for measuring game progress with other methods. When looking at other strong Amazons programs, we see that a recurring theme is the balance between mobility of Amazons in the opening and territory in the endgame. For example, in Lieberum’s program AMAZONG (winner of the Amazons tournament in the 8th Computer Olympiad), the evaluation function is a fine balance between mobility and territory [5]. In this paper, we will compare game progress based on move number with game progress based on mobility, game progress based on territory and game progress based on a combination of these two methods.

In Section 2, we will briefly explain our current method and the problems of using move number for measuring game progress. Section 3 explains the three new methods for measuring game progress. We will also point out some of the problems with these methods. In Section 4, we will give the results of self-play experiments between programs using different methods for measuring game progress. Finally, Section 5 contains the conclusions and suggestions for future work.

## 2 Game Progress Based on Move Number

Because of the large number of possible moves in Amazons, under tournament conditions it is impossible to do a deep full-width search (the full-width search version of our program can only search to a depth of 2 or 3 ply in the opening to 6 or 7 ply near the end of the game). Therefore, using selective search is the only way to do a reasonable look-ahead. The most common domain-

independent methods for selective search, like the null-move heuristic [1], ProbCut [2] and Multi-ProbCut [3] use a shallow search to estimate the result of deep searches. There are two reasons why these methods face problems in Amazons. One reason is that there is no deep search in Amazons until the endgame. Predicting the results of shallow searches by even shallower searches is risky. The second problem is that Amazons programs suffer from the even-odd iteration instability. There are no known methods to do quiescence search in Amazons, so there can be important changes in the evaluation function value after playing a move. A lot of effort into building an evaluation function for Amazons goes into minimizing this effect, but there are still significant differences between the evaluation of even and odd iterations, especially in the opening. This makes it hard to predict the result of a  $d$  ply search with a  $d - 1$  ply search. Also, in the case of shallow searches, the differences between a  $d$  ply search and a  $d - 2$  ply search are usually too large to be useful for a prediction.

Consequently, domain-dependent methods for selective search are needed in Amazons. Eliminating moves from the search is risky, so in our program TAS, we have opted for improving the efficiency of  $\alpha$ - $\beta$  search by searching good moves first. We have proposed a classification of moves based on blocking Amazons and showed that the efficiency of  $\alpha$ - $\beta$  search can be improved significantly (a program with this move ordering scored a 60% winning percentage against a program without any move ordering) [4]. However, we also pointed out that a static move ordering is not the best way to improve search efficiency. In Amazons, like in many other games, the strategic features change from opening to middle game to endgame. Moves that are good in the opening can be decisive mistakes in the endgame and vice versa. Therefore, move ordering should change based on the progress of the game. We used the number of moves from the starting position to change the move order and this further improved the strength of our Amazons program (a program with game progress beat a program without game progress 60% of the time).

However, even though move number is a simple method to measure progress, it is not always correct. As explained in Section 1, a game of Amazons is finished when all territory is fixed and the remaining moves are only about filling the territory, a relatively simple counting problem. The moment when territory becomes fixed is not directly related to move number. Even though territory will be fixed in general around the 50th move, there are often games where the territory becomes fixed much earlier or much later. In these cases, game progress

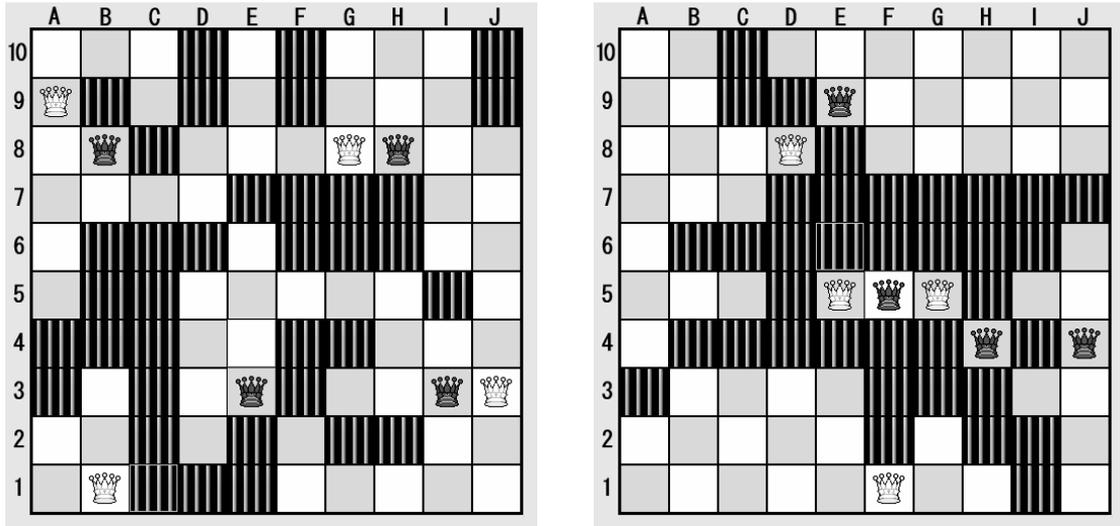


Figure 1: Slow development (left) and fast development (right).

based on move number will give an incorrect value for the game progress and move ordering will suffer.

For example, consider the two positions in Figure 1. Both of the positions in this figure are positions after the 36th move. On the left, there is only one white Amazon trapped inside a small territory, while the territory of the other Amazons is still difficult to decide. In this type of position, trying to trap an opponent Amazon in small territory is the best strategy. This is a strategy that is used in the opening. On the other hand, in the position on the right, the territory of all the Amazons is fixed and the game is finished.

Because of these big differences in game progress despite the same number of moves being played, it is conceivable that measuring game progress based on other methods than move number can further improve the efficiency of the search.

### 3 Other Methods for Measuring Game Progress

Using game progress for attaching weights to evaluation function features has been tried before, particularly in shogi [?]. In shogi, evaluation features that are often used are material, position of pieces relative to the king and king danger. Although as far as we know game progress in shogi is only measured with simple methods, the evaluation function features have a direct relation with game progress. At the start of the game, the material basis is even, while as the game progresses material differences will become larger. Also, in the opening the pieces will be in positions unrelated to either king, while

in the endgame it is important that as many pieces as possible take part in attack or defense. Finally, at the start of the game neither king will be in danger, while near the end of the game at least one of the kings will be under attack by enemy pieces.

In general, evaluation function features are related to the goal of the game, i.e. the final outcome of the game. Low values for features will mean that the goal is far away (the opening stage), while high values mean that the goal is near. Therefore, the progress value is strongly correlated with the features of an evaluation function and these features are therefore a natural choice for measuring progress. We will now explain different ways of measuring progress in Amazons by using evaluation function features that are common in Amazons.

#### 3.1 Territory based game progress

First, in Amazons the concept of territory is an important part of the evaluation function. The territory measurement proposed by Lieberum [5] is a method that might solve the problems with using move number for measuring progress. Even though this method was not proposed as a way of measuring progress, but as a weight to the evaluation function to change the static evaluation into a dynamic one, we feel that this territory measurement is also a good candidate for measuring game progress. The basic idea of using territory as a progress value is that in Amazons the game is over as soon as territory is fixed, i.e. each square is either filled with an arrow, or can only be reached by a single Amazon. Whether a square is going to be a part of the black or white territory will gradually become

clear as the game progresses. In the opening there will be many squares for which it is undecided if they will become black or white territory, while in the endgame there will be many squares for which it is clear if they belong to black or white. The calculation of territory based game progress (TGP) is as follows:

$$\text{TGP} = \sum_a 2^{-|D_W - D_B|}$$

Where  $a$  are the empty squares, and  $D_W$  and  $D_B$  are the minimal number of moves that are needed to move to  $a$  by white and black respectively.

This heuristic is very simple, judging a square as white (black) territory if the number of moves it takes for a white (black) Amazon is lower than the number of moves it takes for a black (white) Amazon to get to this square. The difference between the numbers for white and black is a measurement of the confidence that this square is indeed territory for one side. For example, in the slow development position of Figure 1, consider the squares A6 and D3. In the case of A6, white can move an Amazon to this square in one move, while black needs two moves. Therefore, A6 is considered white territory, but the difference is only one move, so this can change easily and this territory is only weakly decided. On the other hand, in the case of D3, black can move to this square in one move, while white needs three moves, a difference of two moves. In this case, even if white moves one move closer to this square, black still has the time to protect it and it is more likely that D3 will become black territory. In this way, when there is more territory that is decided strongly, the progress value will be higher and the game will be closer to the end.

When we scale TGP to a minimum of 0% (in the starting position) and to a maximum of 100% (in a position with fixed territory), the slow development position of Figure 1 will get a progress value of 51%, while the fast development position gets a value of 100%. Therefore, TGP might be able to solve the problems with progress based on move number.

However, there is also a problem with TGP. This is illustrated by the position in Figure 2. In this position, white has just played D1-G1(D4) and with this move a large number of squares in the lower left corner are almost certain to become white territory. In this case, TGP changes from 30% to 53% in a single move. Sudden changes in the progress value are undesirable, as this leads to instability in move order and therefore to a less efficient search. In positions like Figure 2, TGP fails. A different problem is that when territory is calculated in the evaluation function, only the territory that has changed because of the previous move needs to be recalculated. However, when used for move ordering, this

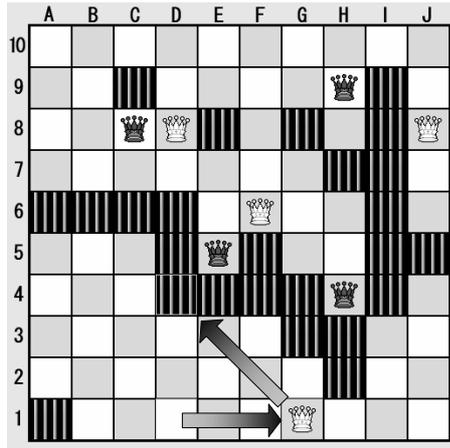


Figure 2: A position where territory based progress fails.

calculation has to be done for every move, which will slow down the search.

### 3.2 Mobility based game progress

A more efficient method for calculating game progress is mobility. This progress value uses the number of times black and white Amazons can go to the same square. In the opening, the mobility of Amazons is high and the number of squares where Amazons of both sides can move to is high. Near the endgame, the mobility of Amazons becomes more and more limited and the number of squares where multiple Amazons can move to becomes smaller and smaller. We use this feature in our second method for calculating progress. The calculation of mobility based game progress (MGP) is as follows:

$$\text{MGP} = \sum_b (M_W + M_B)$$

Here  $b$  are the squares where both black and white can move to, and  $M_W$  and  $M_B$  are the number of white and black Amazons that can move to  $b$  respectively.

As an example, the square E7 in Figure 2 can be reached by the white Amazons on D8, F6 and the black Amazon on E5, so this square gets a value of 3.

When we scale MGP like TGP and set the smallest value to 0% (actually, the starting position has a value of 10%) and the case of fixed territory to 100%, the position of Figure 2 only changes from 74% to 76% after the move D1-G1(D4). Therefore, even if a large number of squares suddenly becomes likely black or white territory, the progress value will not change dramatically, so in this case MGP performs better than TGP.



Table 1: Self-play results for 200 games with 10 seconds per move.

Match			Result	
TGP	-	MN	105	- 95
MGP	-	MN	104	- 96
MTGP	-	MN	95	- 105
MGP	-	TGP	92	- 108
MTGP	-	TGP	93	- 107
MTGP	-	MGP	104	- 96

values are between TGP and MGP values.

## 4 Experimental Results

The three methods for measuring progress that we proposed, like move number, are used to change the order in which moves are searched. To compare these four methods, we performed a number of self-play experiments. For these experiments we have used our Amazons program TAS, which has participated recent Computer Olympiads and has proved that it can play on par with the strongest Amazons programs in the world.

One problem we encountered in Amazons is that it is not trivial to decide how to do self-play experiments. The starting positions should have equal chances to win for both sides, but because there are no experts or opening theory in Amazons, these positions are not easy to find. We generated the initial positions of our self-play experiments by using 100 different positions that were randomly selected from the opening book (after the fourth move from the starting position of the game). These positions were then played twice by each program version, once with white and once with black.

We played a total of 6 matches with 200 games each, giving each program 10 seconds per move. The results of these matches are given in Table 1. The program which measures progress using territory (TGP) wins each match, but the margin of victory is not large enough to draw definite conclusions about playing strength. Even though MTGP was supposed to be the most accurate representation of progress, the results were not as good as expected, which might be caused by the extra computation time that is needed. TGP seems to have the best balance between computation time and accuracy. Another reason might be that accuracy is more important in the opening and middle game, where there are still many moves to choose from. Accuracy in the endgame might not be so important, because there are not so many good moves left.

## 5 Conclusions and Future Work

In this paper, we have proposed three methods for measuring game progress in Amazons. Game progress is used to change the order of moves, aiming at searching good moves first, thus improving  $\alpha$ - $\beta$  search. Our experiments indicate that measuring progress based on territory is the most promising method, even though the results of the self-play experiments were not clear enough for definite conclusions.

In Amazons, game progress can not be used for manipulating the search depth, since the evaluation of a position in Amazons often changes dramatically with a single move. However, there is no reason why game progress should not be used to change search depth in games other than Amazons. In most games, certain moves will have to be investigated deeper in the opening than in the endgame and vice versa. Using game progress in combination with null move pruning, futility pruning and singular extensions is something that we want to try in the future, perhaps in shogi.

Finally, in this research we felt that there are limits to the use of this type of progress measurement. The progress measurements we have proposed are one-dimensional, and we aimed at constructing a progress measurement that is close to a straight line from the starting position (value: 0) to the final position (value: 100). However, in many games, the desired progress value measurement might not be that simple. Instead of a straight line, the desired progress values might form a curve or have some local maxima. To correctly reflect progress in a game, a progress value measurement might need multiple dimensions and this is also an area of future research.

## References

- [1] D. Beal. A Generalised Quiescence Search Algorithm. *Artificial Intelligence*, 43:85–98, 1990.
- [2] M. Buro. ProbCut: An Effective Selective Extension of the alpha-beta Algorithm. *ICCA Journal*, 18(2):71–76, June 1995.
- [3] M. Buro. Experiments with Multi-probcut and a New High-quality Evaluation Function for Othello. In H.J.van den Herik and H.Iida, editors, *Games in AI Research*, pages 77–96. Van Spijk, Venlo, The Netherlands, 2000.
- [4] Y. Higashiuchi and R. Grimbergen. Enhancing Search Efficiency by Using Move Categorization Based on Game Progress in Amazons. In

*Proceedings of the 11th Advances in Computer Games conference*, Taiwan, 2005. (To appear).

- [5] J. Lieberum. An Evaluation Function for the Game of Amazons. In H.J. Van den Herik, H. Iida, and E.A. Heinz, editors, *Advances in Computer Games 10*, pages 299–308. Kluwer Academic Publishers, Boston, USA, 2003.
- [6] H. Matsubara, H. Iida, and R. Grimbergen. Natural developments in game research: From Chess to Shogi to Go. *ICCA Journal*, 19(2):103–112, June 1996.
- [7] N. Sasaki and H. Iida. Report on the First Open Computer-Amazon Championship. *ICCA Journal*, 22(1):41–44, March 1999.
- [8] Wikipedia. [http://en.wikipedia.org/wiki/The\\_Game\\_of\\_the\\_Amazons](http://en.wikipedia.org/wiki/The_Game_of_the_Amazons), 2005.