

## Amazons の評価関数

武田知之

山形大学大学院 理工学研究科

Eメール : dfc43524@dip.yz.yamagata-u.ac.jp

研究で使用していた Amazons のプログラムの中には、局面の判断するに於いての問題点がいくつか存在した。本研究では、その中から3つを選び、研究をした。その3つとは、利きの強さを考慮すること、周りのマスの子数を考慮すること、1つのコマが2つの領域を持つ局面の評価の仕方である。結果は、利きの強さに関してだけ成功した。

## An Evaluation Function for Amazons

Tomoyuki Takeda

Department of Informatics, Yamagata University, Yonezawa, Japan.

E-mail : dfc43524@dip.yz.yamagata-u.ac.jp

In the Amazons program currently used in our research, a number of problems with judging the position existed. In this paper, three of these problems are pointed out and solutions are proposed. The three problems are: not considering the strength of piece coverage, not considering the importance of a surrounding squares and the problem of evaluating a position in which one piece can occupy two areas. The results show that only the proposed solution to the first problem has been successful.

## 1 はじめに

一般的に、与えられた局面に存在する可能な手の数が増えると深く探索するのは難しくなる。それによって、局面の途中で局面を評価することの重要性が増す。つまり、平均の手の数が多いゲームほど評価関数が重要となる。平均の手の数は、ゲームによって異なる。例えば、将棋は80手、チェスは35手と違う[1]。2つの例でいうと、チェスよりも将棋のほうが深く局面を読むのは難しい。本研究で用いる Amazons においては、平均の手の数が479手[1]と多い。つまり、チェスや将棋よりも評価関数が重要となってくるのである。

第2章では、研究するにあたって必要な知識について説明する。第3章では、Amazons のプログラムにある評価関数の問題点をあげ、それをどのように改善するかを説明する。第4章では、改良版プログラムが実際に成果のあるものかを見るための実験の結果を説明する。第5章では、まとめと今後の課題について記述する。

## 2 必要な知識

### 2.1 Amazons のルール

ここで、Amazons のルールについて説明する。Amazons では、白・黒それぞれ4つのコマを持つ。初期局面は10×10

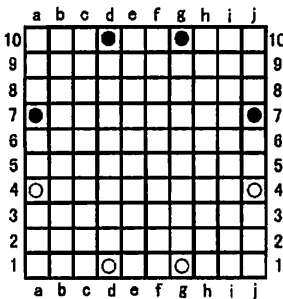


図1 初期局面

の盤上に図1のようにコマを配置する。ゲ

ームは白の手番から始めて、白・黒交互に4つのコマのうちの一つを動かして進めていく。

コマの行動には2つある。まず、コマを空いているマスに移動する。その後矢を投

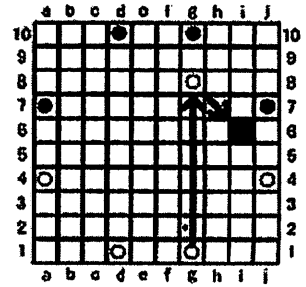


図2 2つの行動

げるといふ行動である。行動範囲は、両方ともチェスのクイーンの動きと同じである。また、矢が投げられた場所やコマの上は通過できない。行動の例を図2に示す。

最後に勝敗について説明する。こうした行動を繰り返すうちに、矢によってコマが移動できるところが限られてくる。そして、ど

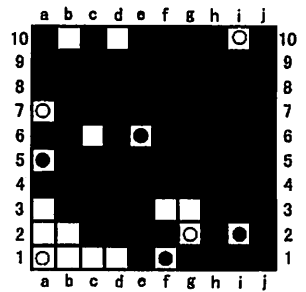


図3 終了局面

のコマも移動するところなくなったほうが負けとなる。図3では、黒の手番であるが、動かせるコマがないので、白の勝ちである。

### 2.2 局面の有利・不利

Amazons の有利・不利を判断には、主に「可動性」と呼ばれるものと「領域」と呼ばれる2つの要素を使う。

### 2. 2. 1 可動性

可動性とは、コマがどれだけ動けるかを表すものである。周りに矢やコマが少なければ可動性は上がる。序盤はできるだけ、可動性が高くなるようにコマを動かすことが重要となる。

### 2. 2. 2 領域

領域とは、白または黒のコマだけが移動可能なマスのことである。図4の点線の入っているマスが白の領域である。

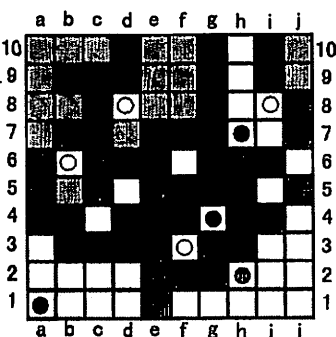


図4 白の領域

また、領域となりそうなマスに「潜在的領域」と呼ぶ。もしも、白のほうが黒よりも少ない手数であるマスに到達できるとするとそのマスは、白の潜在的領域となる。図5の網線の入っているマスが白の潜在的領域である。

Amazonsの勝敗から、自分だけが移動できるマスを多く持っていたほうが有利となる。つまり、より多くの領域を持っていたほうが有利と判断できる。

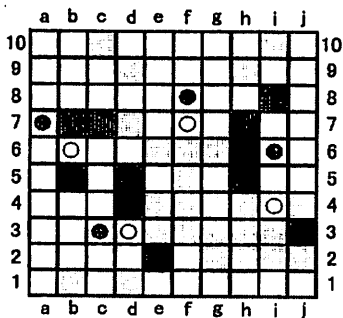


図5 白の潜在的領域

### 3 3つの問題点と改善方法

#### 3. 1 利きの強さ

「利き」とは、あるマスにコマが1手でいくことができるかを表すものである。また、「利きの強さ」とはあるマスに対して利きのあるコマの数を表す。

図6の網線のマスは、白も黒も利きがある。したがって、点数としては0である。つまり、どちらもそのマスにおいては優位性がないということである。しかし、利きの強さをみると白のほうが多い。よって、網線のマスは白のほうの優位性が高いと判断できる。

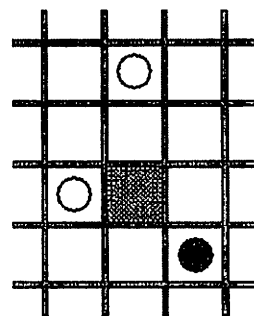


図6 利きの強さ

そこで、あるマスの白と黒の利きの強さの差を計算する。そして、その計算結果に重みをつけることで点数を与える関数を新しく定義し改善を図る。式は以下に示す。

$$K = \sum_{T(a)} (W \times S(a))$$

Wは重み、S(a)はマスaの白と黒の利きの強さの差をKは利きの点数を表す。また、T(a)は以下の条件である。

条件 T(a): aが空きマスで白と黒両方の利きがある

この改良したプログラムをKSと呼ぶこととする。

#### 3. 2 周りのマスの必要性

図7の網線のマスだけを見ると白よりに高い点数がつく。しかし、網線のマスの周りの点線の8マスを考えると黒のほうに点

数が傾いている。網線のマスは、黒の潜在的領域にほぼ囲まれているので、先を見越すと白の領域となる可能性は低い。

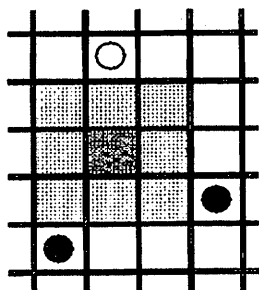


図7 周りのマス

今のままでは、そこまで計算することはできない。したがって、あるマスに点数をつけるときは周りのマスも考慮するように変更する。方法として、まず周りのマスを考慮しないですべてのマスに点数をつける。その後、点数をつけたいマスの周りのマスの点数の平均値をとって加えるというふうに変更する。式は以下のとおりである。

$$A = \sum_{\text{emptysquares } a} \left( M(a) + R(a) \times \frac{N}{10} \right)$$

$M(a)$ はマス  $a$  の中央のマスの点数、 $R(a)$ はマス  $a$  の周りのマスの点数の平均値を  $A$  は周りのマスの点数を表す。また、 $N$ は任意の整数を表す。この改良したプログラムを  $AS$  と呼ぶこととする。

### 3.3 1つのコマが多数の領域を持つ

図8-aの局面では、f8の白のコマは

点線の領域と網線の領域を持つ。しかし、実際は点線の領域か網線の領域どちらかしか所持すること

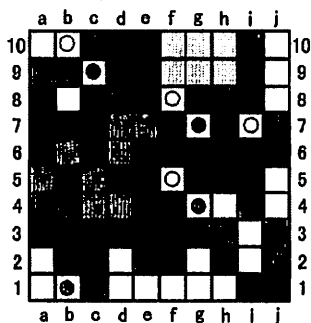


図8-a コマ1領域2

はできない。例えば、f8のコマが点線の領域に移動し、f8に矢を放つと図8-b

のようになり、網線のマスにはいけなくなる。矢を放たなければ、図8-cのようになり、g7の黒のコマで網線の領域は

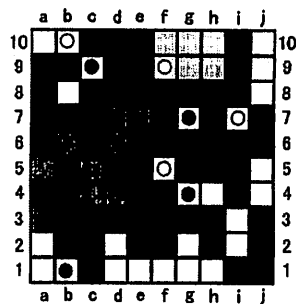


図8-b 1手目の例1

取られてしまう。逆に網線の領域に移動したとしても、同様のことが言える。

そこで、所持している領域の数を比較し、大きい方だけを領域として数えるようにプログラムに変更を加える。

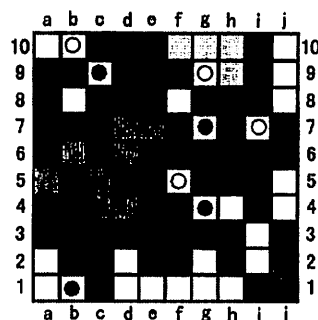


図8-c 1手目の例2

例えば、図8-aの局面では網線の領域の数が8、点線の領域の数が6で網線の領域の数の方が大きいので、f8のコマは網線の領域つまり8個の領域を持っていることとなる。この改良したプログラムを  $ME$  と呼ぶこととする。

## 4 実験方法と実験結果

### 4.1 実験方法

KSは、重み  $W$  の値を0.05刻みで増やしていき、元のプログラムと50戦させる。また、必要であれば追加実験を行いより正確なデータを取るようにした。

ASは、整数値N1の値を1～10まで変化させていき、元のプログラムと50戦させる。実験結果がいいものやあまりにも勝敗が変動したものについては、追加実験を行い、より正確なデータを取るようにした。

MEは、テスト局面1～4を用意し、その局面の白をコンピュータ、黒を自分で動かして白が勝てるかを実験する。テスト局面は、図8-aのように1つのコマが領域を2つ以上保持している局面が生じるものを考え、その中考えられる形をそろえたものである。また、その後には元のプログラムと100戦し、全体的な成果はどうなっているのかを見た。

#### 4.2 実験結果

3つの改良したプログラムの最も結果のよかったデータについて100戦させて、より正確なデータにしたものを表1に示す。プログラム名に書いてある括弧の中は変化させた値がいくつときのデータかを表す。

表1 3つの100戦結果

プログラム名	勝敗
KS-元のプログラム (W=0.17)	60-40
AS-元のプログラム (N1=8)	50-50
ME-元のプログラム	10-90

結果より、KSについてだけ効果があった。ASについては、中盤辺りから周りの空きマスの数が減ったために改良の意味がなくなると考えられる。また、MEにつ

いては改良の際に点数をいじってしまったので、点数のバランスが崩れてしまったためにこんな結果となったと考えられる。

また、テスト局面1～4についてMEを用いて行った実験結果を表2に示す。表の見方は、○が勝つことができたかを表し、×が9手先探索にしても勝てなかったことを表す。また、括弧内に書いてあるのは深さいくつから勝てたかを表している。

表2 テスト局面を用いた実験結果

局面名	ME(深さ)	元のプログラム
テスト局面1	○(1)	×
テスト局面2	○(1)	×
テスト局面3	○(2)	×
テスト局面4	○(2)	×

表2の結果より、MEでは元のプログラムでできなかったことができるようになった。しかし、深さが2必要な局面が存在している。これは、領域より潜在的領域を優先させてしまった結果と考えられる。

つまりMEは部分的には改善できているが、全体的にみると逆効果であったと言える。

#### 5 まとめと今後の課題

本研究では、評価関数の問題点である3つのことについて研究してきた。1つ目は、利きの強さを考慮に入れること。2つ目は、中央のマスの点数を決めるのに周りのマスの点数も考慮に入れること。3つ目は、コマの1つが2つ以上の領域を持っている局

面で正しく判断させることである。

それぞれに対して、プログラムに改良を加えた。利きの強さに関するものには、利きの数を数える関数と利きの強さを計算し、そこから点数を計算する関数を定義した。周りのマスの点数を考慮に入れることに関しては、中央のマスと周りのマスの点数を対比させ、それにより最終的な点数をつける方法を取り入れた。また、コマの1つが2つ以上の領域を持つ局面の判断では、1つのコマがそのコマだけが保持している領域が2つ以上あった場合、領域の数を計算し小さいほうの領域を削除し、結果的に大きいほうの領域だけを計算できるように改良を加えた。

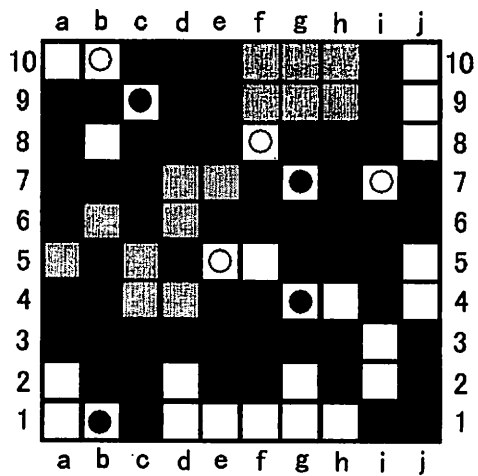
利きの強さについての改良は効果があった。また、周りのマスを考慮に入れることに関する改良については、効果がなかった。最後のコマ1つが2つ以上の領域を持つ局面の判断についての改良は、テスト局面を用いた実験では成果があったが、実際に対戦させると逆にマイナスの効果しか生まなかった。

今後の課題としては、2手以上先の利きについてを考慮すること、周りのマスの数を場合によって変化させていくとどうなるか。また、点数のつけ方を平均点ではなくて、1マス1マスを考えて点数をつけるようにするとどうなるかということ。序盤・中盤・終盤で、可動性や領域、潜在的領域がどれだけの影響を与えるのかを研究する必要がある。

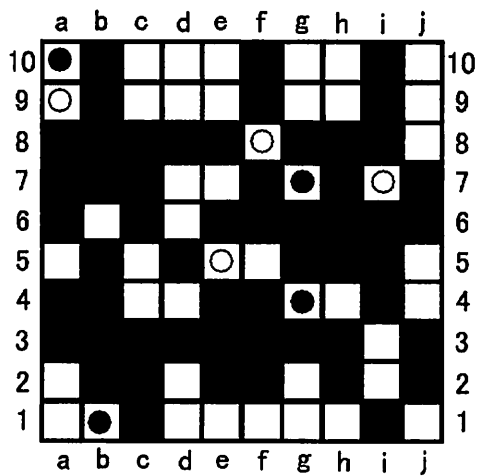
## 参考文献

- [1] Higashiuchi, Y. and Grimbergen, R. Enhancing Search Efficiency by Using Move Categorization Based on Game Progress in Amazons. Proceedings of the 11th Advances in Computer Games conference, Taiwan (to appear) (2005)

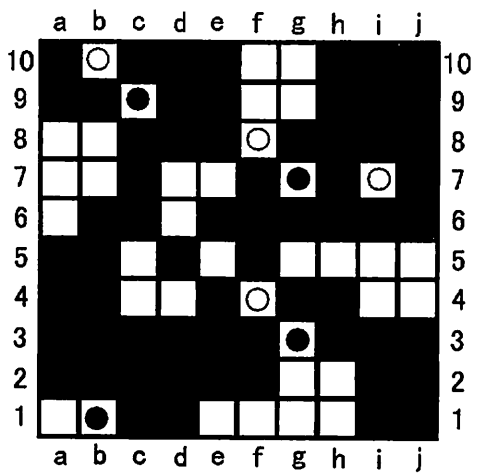
# 付録



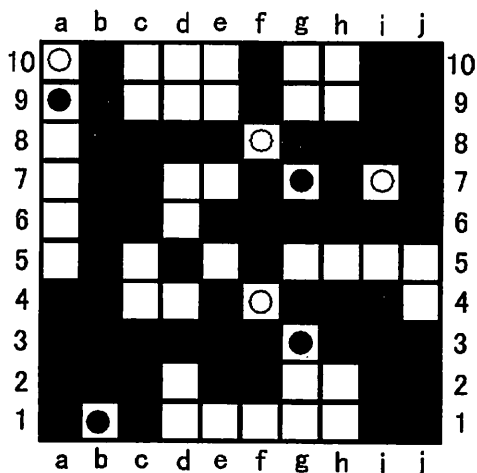
付録1 テスト局面1



付録2 テスト局面2



付録3 テスト局面3



付録4 テスト局面4