

情報論理工学 研究室

第7回： 強い手の選択



1

ゲームAIの作成

- ゲームAI作成には何が必要か？
 - ルール通りに指せる・打てる
 - 合法手の中で強い手を選べる
 - プレイヤーの手が合法手か判定できる
 - 合法手を指した・打った後の局面を生成できる
 - 終了判定ができる
 - 得点計算・勝敗判定ができる

2

「強い手」の選択

- 「強い手」の選択
 - 強い手とは？
 - 大きな得点が得られる手
 - 相手の得点を下げる手
 - 価値の高い駒を取る手
 - 価値の高い駒を守る手
 - 有利な地点を取る手
 - 相手に不利な地点を取らせる手
 - 有利な選択ができるようになる手
 - 相手に不利な選択を強要する手

ゲームによって異なる

3

「強い手」を得る手法

- 「強い手」を得る手法
 - 局面の評価値計算
 - 定跡・定石データベースの利用
 - 先読み
 - 完全読み切り・必勝読み切り
 - モンテカルロ法
 - 機械学習

4

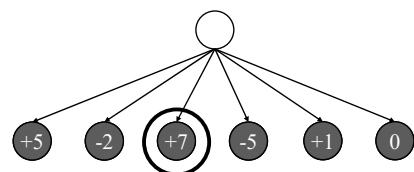
局面の評価値計算

- 局面の評価値計算
 - 現在の局面からどのくらい優勢かを計算する
 - 得点を多く取っている
 - 盤上に強い駒がある
 - 強いカードを持っている
 - 有利な地点を抑えている
 - 相手を攻撃できる
 - 相手の攻撃を防げる
 - 可能な手の数が多い

5

評価値計算による手の選択

1. 各合法手に対する1手先の局面を生成
2. 各局面の評価値を計算
3. 最も評価値の高い手を選択



6

```

/* 最も評価値の高い手を選ぶ */
Move SelectMove () {
    ArrayList<Move> moveList = generateMoveList (); // 合法手リスト生成
    if (moveList.isEmpty()) return null; // 合法手無し
    int selectedValue = -∞; // 局面の評価値
    Move selectedMove = null; // 選択した手
    for (Move move : moveList) {
        Phase phase = nextPhase (move); // 1手先の局面を生成
        int value = phase.getValue(); // 局面の評価値を求める
        if (value > selectedValue) { // 評価値最大の手を記憶
            selectedMove = move; selectedValue = value;
        }
    }
    return move;
}

```

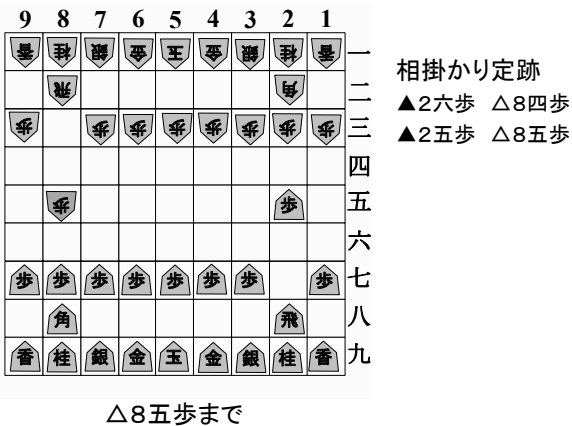
7

定跡・定石データベース

- 定跡・定石
 - 特定の局面で一般に強いとされている手
 - 序盤定跡・序盤定石
 - 中盤定跡・中盤定石
 - 終盤定跡・終盤定石
- 定跡・定石データベース
 - 各局面でプロが指した手のデータベース
 - 定跡にある局面では定跡通りに指す

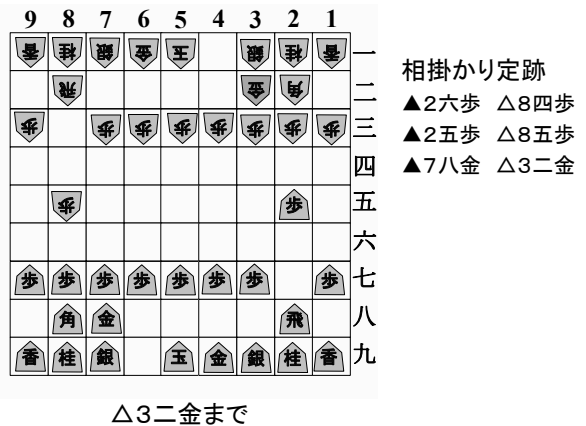
8

定跡の例:将棋



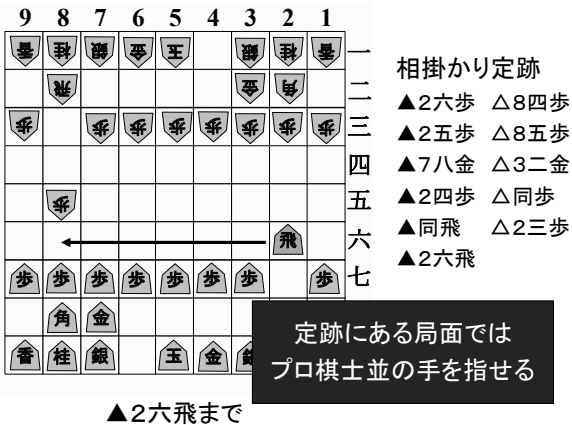
9

定跡の例:将棋



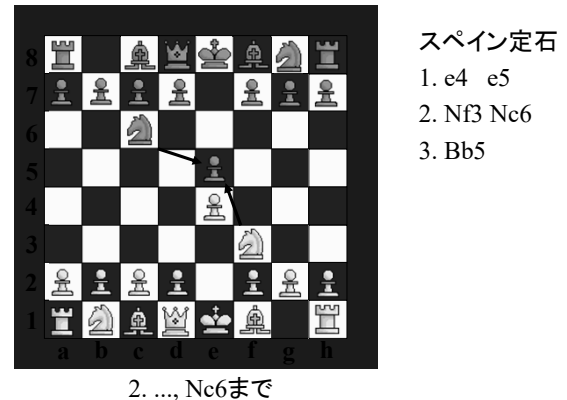
10

定跡の例:将棋



11

定跡の例:チェス



12

定跡の例:チェス



- スペイン定石
 1. e4 e5
 2. Nf3 Nc6
 3. Bb5

3. Bb5 まで

13

定跡の例:チェス



- スペイン定石
 1. e4 e5
 2. Nf3 Nc6
 3. Bb5 a6

3. ..., a6 まで

14

定跡の例:チェス

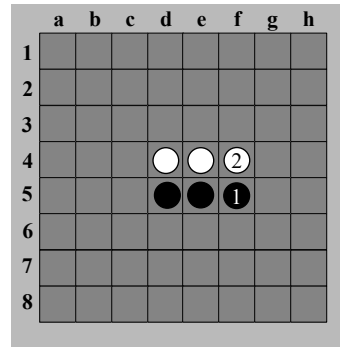


- スペイン定石
 1. e4 e5
 2. Nf3 Nc6
 3. Bb5 a6
 4. Ba4 Nf6
 5. O-O

5. O-O まで

15

定石の例:リバーシ

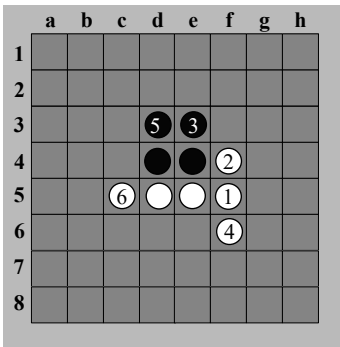


- 鼠定石
 1:黒f5 2:白f4

2:白f4 まで

16

定石の例:リバーシ

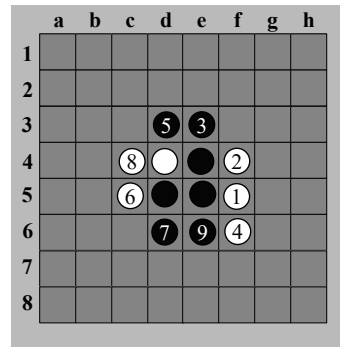


- 鼠定石
 1:黒f5 2:白f4
 3:黒e3 4:白f6
 5:黒d3 6:白c5

6:白c5 まで

17

定石の例:リバーシ

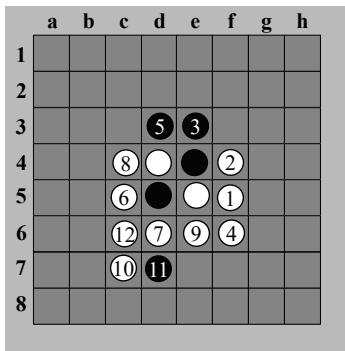


- 鼠定石
 1:黒f5 2:白f4
 3:黒e3 4:白f6
 5:黒d3 6:白c5
 7:黒d6 8:白c4
 9:黒e6

9:黒e6 まで

18

定石の例:リバーシ



- 鼠定石
- 1:黒f5 2:白f4
 - 3:黒e3 4:白f6
 - 5:黒d3 6:白c5
 - 7:黒d6 8:白c4
 - 9:黒e6 10:白c7
 - 11:黒d7 12:白c6

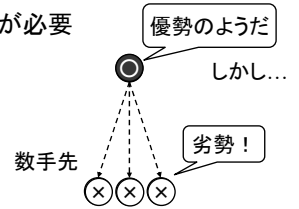
12:白c6 まで

19

先読み

現在の局面は優勢でも
数手先に逆転される場合がある

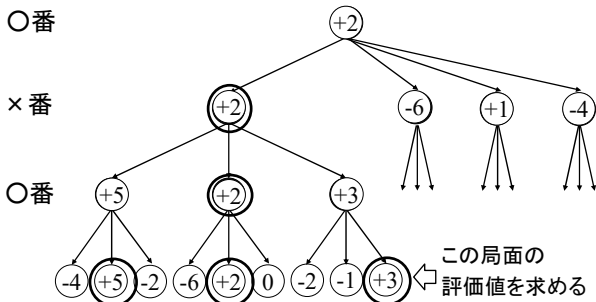
↓
局面の先読みが必要



20

先読み

- 先読み
 - 数手先の局面を生成し、評価値を計算



21

可能な局面数

		各局面での合法手数		
		2	4	8
先読み手数	10	1,000	1,000,000	10^9
	20	1,000,000	10^{12}	10^{18}
	30	10^9	10^{18}	10^{27}
	40	10^{12}	10^{24}	10^{36}
	50	10^{15}	10^{30}	10^{45}

先読み手数が増えると可能な局面数は指数的に増える
⇒ 適当な枝刈りが必要

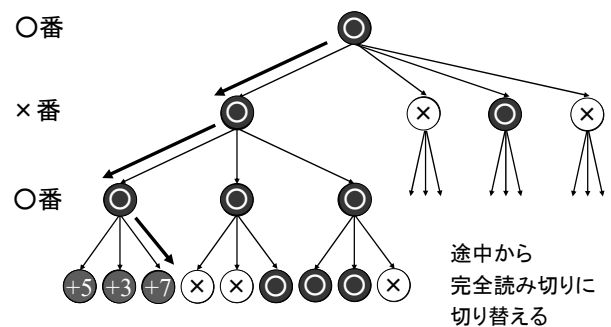
22

完全読み切り・必勝読み切り

- 完全読み切り
 - ゲーム終了まで読む
 - 最も得点が高い勝つ手を選択する
 - 必勝読み切り
 - ゲーム終了まで読む
 - 勝つ手を選択する
- ゲーム終盤で用いる
まず必勝読みで勝利を確定し、
さらに完全読みで得点の高い勝ちを目指す

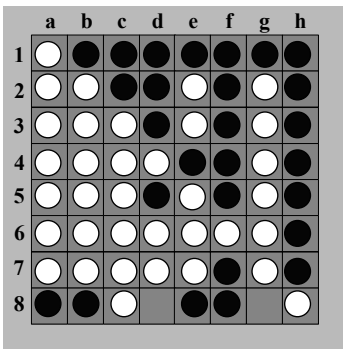
23

完全読み切り・必勝読み切り



24

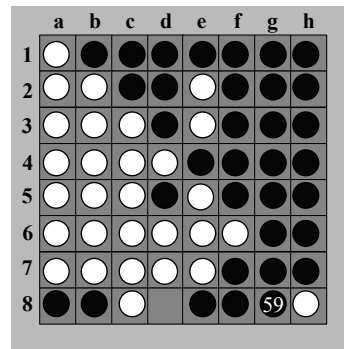
完全読み切りの例:リバーシ



黒番
d8 と g8
どちらに打つ？

25

完全読み切りの例:リバーシ



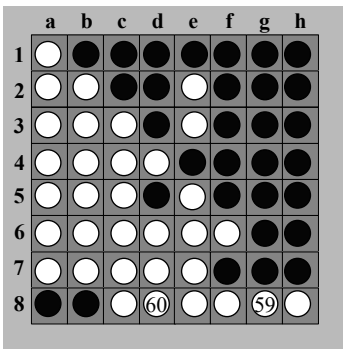
白番
g8に打つと...

59: 黒g8

59:黒g8 まで

26

完全読み切りの例:リバーシ



g8に打つと...

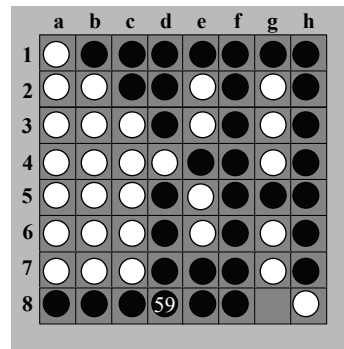
59: 黒g8 60: 白d8

黒31对白33
白の勝ち

60:白d8 まで

27

完全読み切りの例:リバーシ



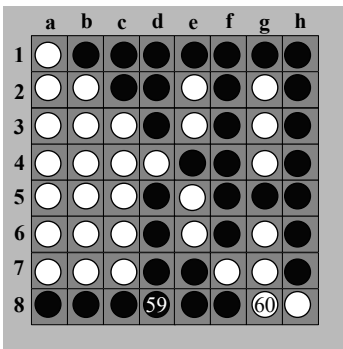
白番
d8に打つと...

59: 黒d8

59:黒d8 まで

28

完全読み切りの例:リバーシ



d8に打つと...

59: 黒d8 60: 白g8

黒33对白31
黒の勝ち

60:白g8 まで

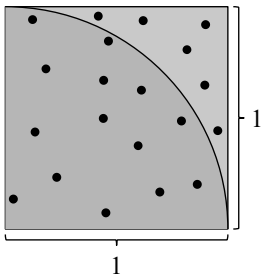
29

モンテカルロ法

- モンテカルロ法
 - ランダムアルゴリズム
 - 多項式時間で答が出るが、正しいとは限らない
- ⇔ラスベガス法
 - ランダムアルゴリズム
 - 答が出るとは限らないが、出た場合は正しい答

30

モンテカルロ法の例:円周率



1. 正方形の中にランダムに点をたくさん打つ
2. 扇型内の数/全体の数を求める

扇型内:14個
全体 :20個

$14/20 = 0.7$
 $0.7 * 4 = 2.8$

⇒円周率 ≒ 2.8

扇型部分の面積は

$1 * 1 * \pi / 4$

点の数を増やせば3.14に近づくはず...

31

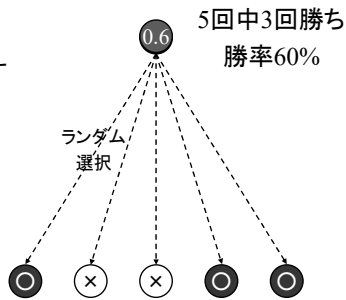
/* モンテカルロ法で円周率を計算する */

```
Double ComputePi() {
    final int IMAX = 10000; // 反復回数
    long seed = System.currentTimeMillis(); // 現在時刻を得る
    Random rnd = new Random (seed); // 乱数生成
    int inAsector = 0;
    for (int i=0; i<IMAX; ++i) {
        Double x = rnd.nextDouble(); // 0~1の乱数を生成
        Double y = rnd.nextDouble();
        if ((x*x + y*y) < 1) ++inAsector; // 扇型内に入った数をカウント
    }
    Double pi = (Double) inAsector / (Double) IMAX * 4.0;
    return pi;
}
```

32

モンテカルロ法

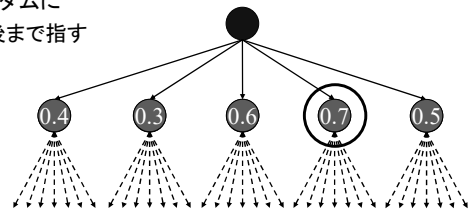
ランダムに最後まで指す
(敵味方とも)



33

モンテカルロ法

各手に対してランダムに最後まで指す



最も勝率の高い手を選択

34

局面の評価値計算(再掲)

■ 局面の評価値計算

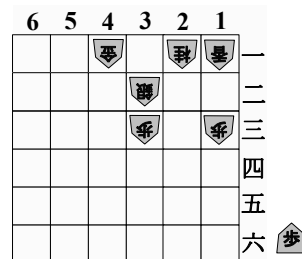
- 現在の局面からどのくらい優勢かを計算する
 - 得点を多く取っている
 - 盤上に強い駒がある
 - 強いカードを持っている
 - 有利な地点を抑えている
 - 相手を攻撃できる
 - 相手の攻撃を防げる
 - 可能な手の数が多い

35

駒割り

■ 駒割り(駒の損得勘定)

- 弱い自駒と強い敵駒を交換できれば有利



▲2二歩と打てば
2一の柱と交換できる

36

駒の価値：将棋

駒の価値

歩	香	桂	銀	金	角	飛	玉
1	5	6	8	9	13	15	∞
と	杏	圭	全		馬	龍	
12	10	10	9		15	17	

と杏圭は動きは金と同じだが、敵に取られても渡すのは歩香桂ですむので金よりも価値が高い

谷川17世名人による

37

駒割りによる評価値の例

	9	8	7	6	5	4
一				王	王	
二		飛				
三				飛	飛	
四						
五						角
六						桂

ここで駒得するには？

38

駒割りによる評価値の例

▲7三角

	9	8	7	6	5	4
一				王	王	
二		飛				
三				角	飛	飛
四						
五						
六						桂

▲7三角まで

7三角で
王手飛車！

飛車(15点)を
必ず取れる！

王手飛車！

39

駒割りによる評価値の例

▲7三角 △6二飛

	9	8	7	6	5	4
一				王	王	
二				飛		
三				角	飛	飛
四						
五						
六						桂

△6二飛まで

ここで
▲6二同角とすれば
飛車を取れる

だがそれでは
飛車(15点)と
角(13点)の交換

40

駒割りによる評価値の例

▲7三角 △6二飛
▲7四桂

	9	8	7	6	5	4
一				王	王	
二				飛		
三				角	飛	飛
四				桂		
五						
六						

▲7四桂まで

▲7四桂で
飛車(15点)と
桂(6点)の交換

41

駒の価値：チェス

駒の価値

兵	馬	象	車	王	后
1	3	3	5	9	∞/4(※)

(※) キングの価値は∞だが、戦力として見た場合は4 (将棋の玉将と異なりチェスではキングも攻め駒として使われる)

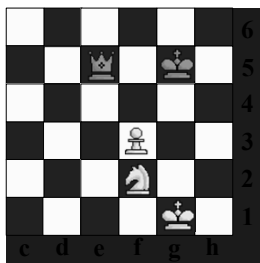
6段目	7段目
3	5

ポーンは8段目まで進むと任意の駒になれるので、6,7段目まで来たポーンは価値が上がる

Bobby Fisher による

42

駒割りによる評価値:チェス



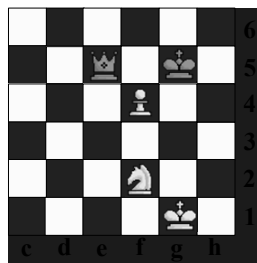
先手:4点
後手:9点

後手の方が
有利だが...

43

駒割りによる評価値:チェス

1. f4+



先手:4点
後手:9点

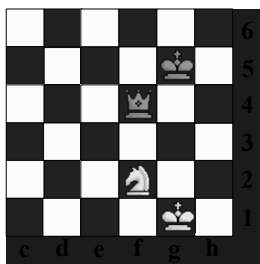
ポーンでチェック!

1.f4+ まで

44

駒割りによる評価値:チェス

1. f4+ Qxf4



先手:3点
後手:9点

クイーンで取るが
...

1.f4+ Qxf4 まで

45

駒割りによる評価値:チェス

1. f4+ Qxf4

2. Nh3+

先手:3点
後手:9点

ナイトで両取り!

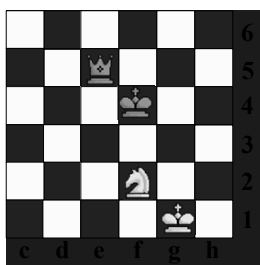
クイーン(9点)と
ナイトポーン(4点)の交換

2. Nh3+ まで

46

駒割りによる評価値:チェス

1. f4+ Kxf4



先手:3点
後手:9点

キングで取っても...

1.f4+ Kxf4 まで

47

駒割りによる評価値:チェス

1. f4+ Kxf4

2. Nd3+

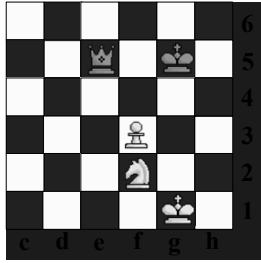
先手:3点
後手:9点

やっぱり
ナイトで両取り!

2. Nd3+ まで

48

駒割りによる評価値:チェス



先手:4点
後手:9点

後手の方が
有利だが...

クイーン(9点)と
ナイトポーン(4点)の交換

49

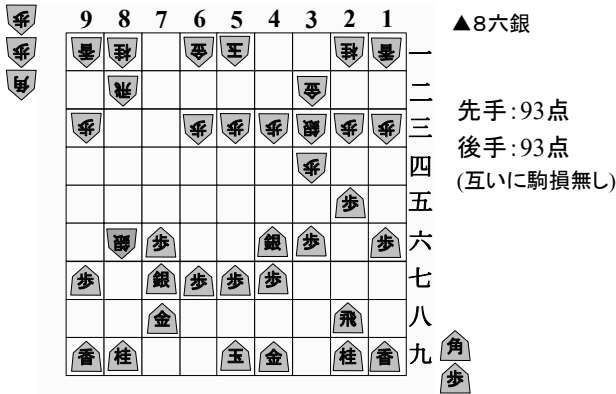
駒割りによる評価値計算

- 駒割りによる評価値計算
 - 強い駒を取れる手を高評価にする

これである程度はいい手を選べるが...

50

駒割による評価値:将棋



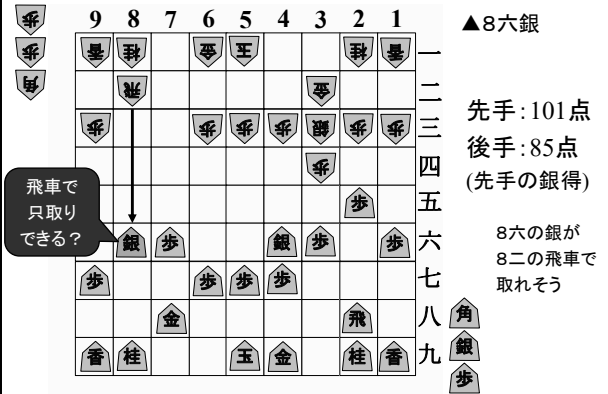
▲8六銀

先手:93点
後手:93点
(互いに駒損無し)

△8六銀まで

51

駒割による評価値:将棋



▲8六銀

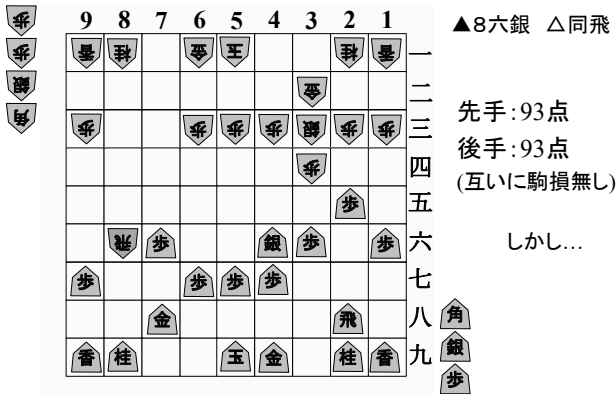
先手:101点
後手:85点
(先手の銀得)

8六の銀が
8二の飛車で
取れそう

▲8六同銀まで

52

駒割による評価値:将棋



▲8六銀 △同飛

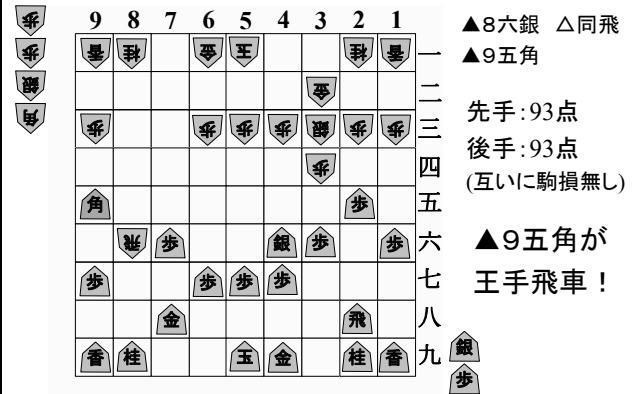
先手:93点
後手:93点
(互いに駒損無し)

しかし...

△8六同飛まで

53

駒割による評価値:将棋



▲8六銀 △同飛

▲9五角

先手:93点
後手:93点
(互いに駒損無し)

▲9五角が
王手飛車!

▲9五角まで

54

駒割りによる評価値: チェス



4. ... Bg4

先手: 39点
後手: 39点
(互いに駒損無し)

4. ... Bg4 まで

55

駒割りによる評価値: チェス



4. ... Bg4
5. Nxe5!?

先手: 39点
後手: 38点
(先手のポーン得)
d1のクイーンが
g4のビショップで
取れそう

クイーンを取れる?
5. Nxe5 まで

56

駒割りによる評価値: チェス



4. ... Bg4
5. Nxe5!? Bxd1

先手: 30点
後手: 38点
(後手のクイーン得)

しかし...

5. Nxe5 Bxd1 まで

57

駒割りによる評価値: チェス



4. ... Bg4
5. Nxe5!? Bxd1
6. Bxf7+

先手: 30点
後手: 37点
(後手のクイーン得)

得点はまだ
黒が勝っているが...

6. Bxf7+ まで

58

駒割りによる評価値: チェス



4. ... Bg4
5. Nxe5!? Bxd1
6. Bxf7+ Ke7

先手: 30点
後手: 37点
(後手のクイーン得)

6. Bxf7+ Ke7 まで

59

駒割りによる評価値: チェス



4. ... Bg4
5. Nxe5!? Bxd1
6. Bxf7+ Ke7
7. Nd5#

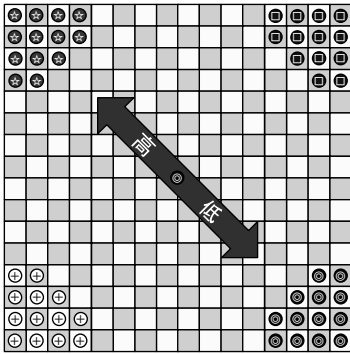
先手: 30点
後手: 37点
(後手のクイーン得)

この状況では
駒割り計算は
意味無し

7. Nd5# まで

60

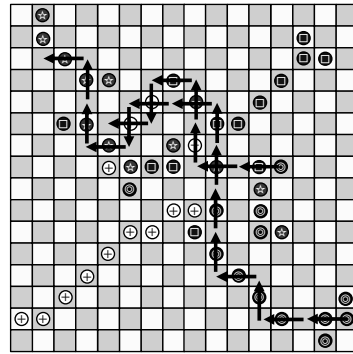
位置による評価値:ハルマ



ゴールに近い駒を
高評価にする
⇒ゴールに近づく手を
高い評価にする

67

位置による評価値:ハルマ



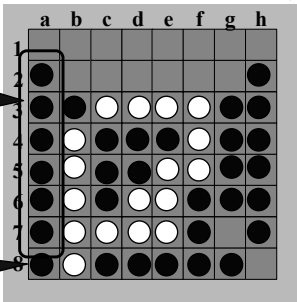
できるだけゴールに
近づく手を選択

68

位置による評価値:リバーシ

- リバーシの有利な位置
 - 確定石(ゲーム終了までひっくり返せない)

隅から繋がる辺の石も
引っ繰り返せない



確定石が
多いと
有利!

隅の石は
引っ繰り返せない

69

位置による評価値:リバーシ

	a	b	c	d	e	f	g	h
1	隅	C	A	B	B	A	C	隅
2	C	X					X	C
3	A							A
4	B							B
5	B							B
6	A							A
7	C	X					X	C
8	隅	C	B	B	B	A	C	隅

隅のマスは取ると
確定石
⇒隅を取ると有利

隅の隣接マス
(X,C)は取ると
相手に隅を
取られやすい
⇒X,Cマスを取ると不利

70

位置による評価値:リバーシ

	a	b	c	d	e	f	g	h
1	10	-5	2	0	0	2	-5	10
2	-5	-7	-1	-2	-2	-1	-7	-5
3	2	-1	-2	-1	-1	-2	-1	2
4	0	-2	-1			-1	-2	0
5	0	-2	-1			-1	-2	0
6	2	-1	-2	-1	-1	-2	-1	2
7	-5	-7	-1	-2	-2	-1	-7	-5
8	10	-5	2	0	0	2	-5	10

有利なマスに正
不利なマスに負の
値を割り当てる

得点の高い
マスに打つ

これだけでも
ある程度の強さにはなる

各マスへの得点割り当ての例

71

着手可能手数

- 着手可能手数(その局面で指せる手の数)
 - 着手可能手数が少ない
⇒不利な手でも選ばないといけない

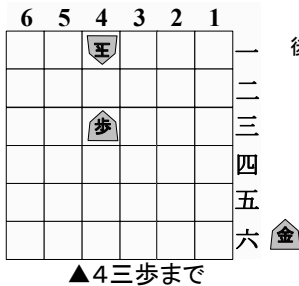
自分の着手可能手数が多く
相手の着手可能手数が少ない手を
高評価に

72

着手可能手数の例:将棋

■ 詰めろ

- 適切な対処をしないと詰む状態



後手玉は▲4二金までの詰めろ

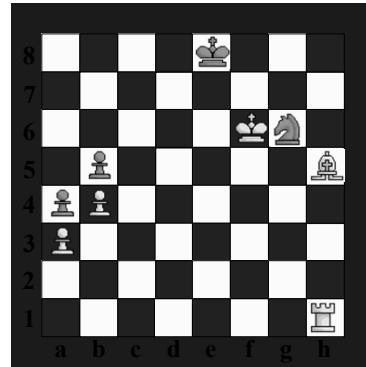
⇒▲4二金を防ぐ手しか指せない

▲4三歩まで

73

着手可能手数の例:チェス

白番



74

着手可能手数の例:チェス

黒番

1. Rd1

ポーンは動けない

ビショップがいるのでナイトは動けない

黒の着手可能手は Kf8 のみ

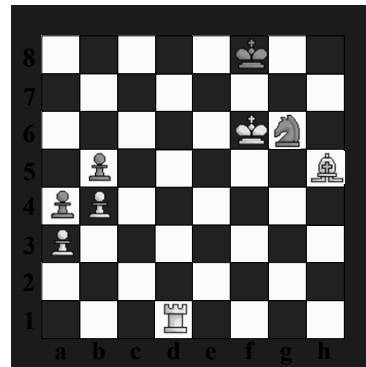
1. Rd1 まで

75

着手可能手数の例:チェス

白番

1. Rd1 Kf8



1. Rd1 Kf8 まで

76

着手可能手数の例:チェス

黒番

1. Rd1 Kf8

2. Rd8#

Kf8 以外の手をもし指せていれば詰まなかった...

仮にパスができたとしたら詰んでいない

チェックメイト!

2. Rd8# まで

77

着手可能手数

- 着手可能手数(その局面で指せる手の数)

- 着手可能手数が少ない

⇒不利な手でも選ばないといけない

自分の着手可能手数が多く相手の着手可能手数が少ない手を高評価に

しかし場合によっては相手の着手可能手数を増やすことも

78

勝負手

着手可能手数が少ない

⇒ 読まなければならない手が少ない

⇒ 先を読みやすい

着手可能手数が多い

⇒ 読まなければならない手が多い

⇒ 迷い・判断ミスが起きやすい

不利なときはあえて相手の選択肢を増やして

相手のミスを誘う

勝負手

機械学習

■ 機械学習

■ 対戦データから「勝つための方法」を学習

■ 深層学習 (deep learning)

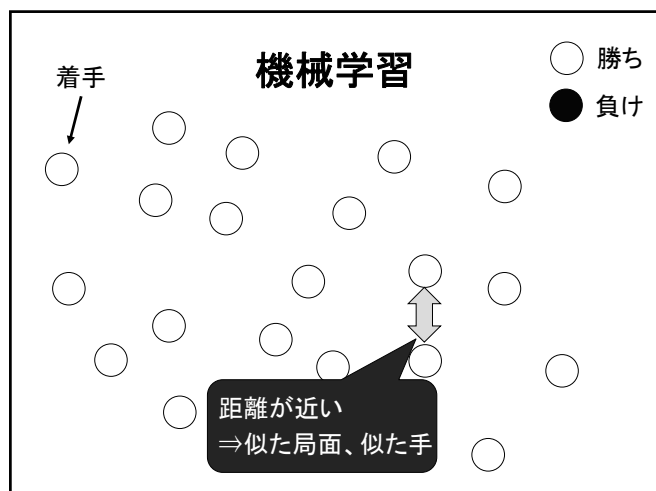
■ 機械学習の発展型

対戦データを分析して

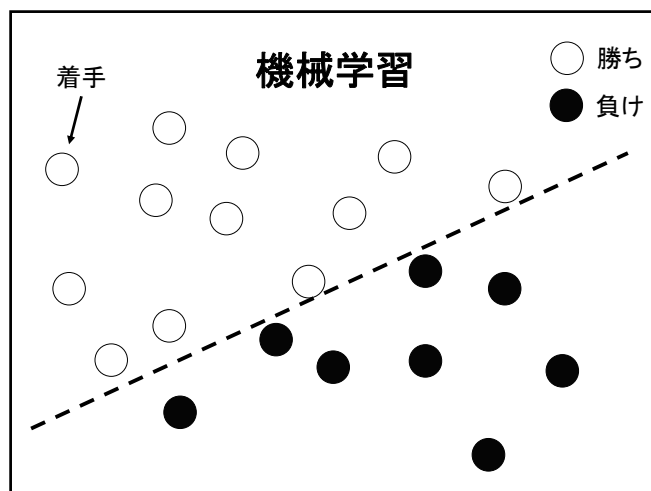
「勝つパターン」と「負けるパターン」に分類する

79

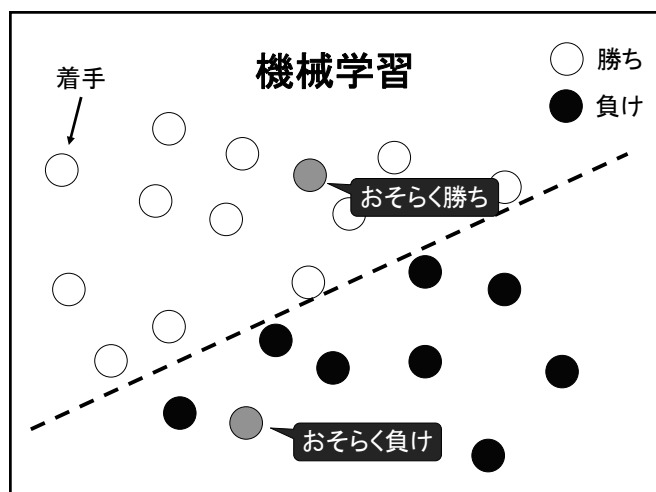
80



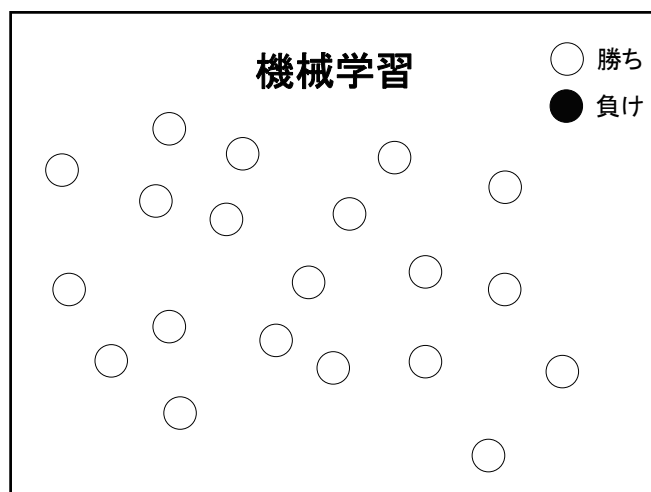
81



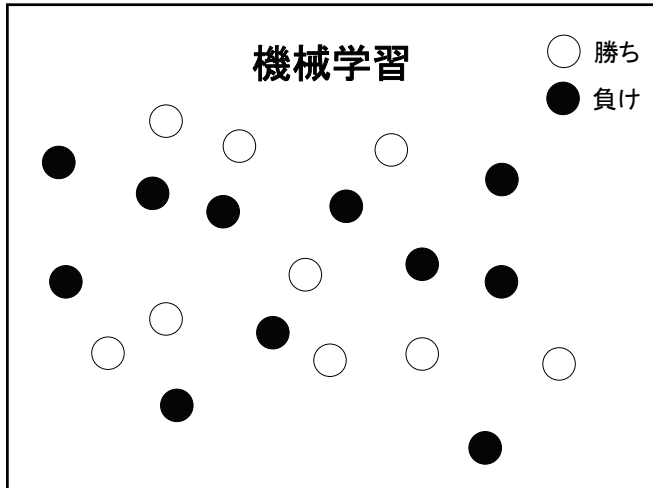
82



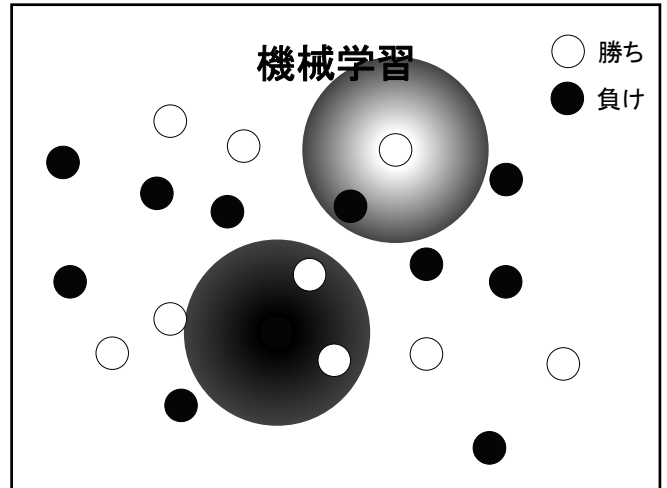
83



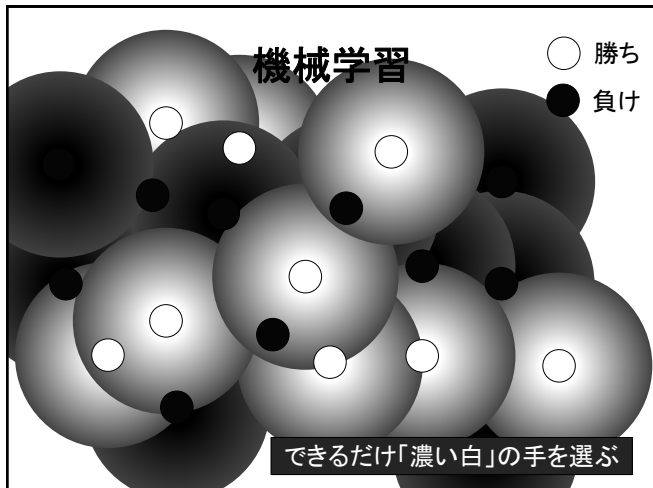
84



85



86



87

宿題:3目並べの着手選択

- 3目並べ着手選択
 - 先手後手それぞれを人間かCPUのどちらが受け持つかを選べるようにせよ
 - CPUは以下の手順で着手決定
 - 自分にリーチが掛っている場合
 - ⇒ 勝つ手を打つ
 - 自分にリーチが掛っておらず、相手にリーチが掛っている場合
 - ⇒ それを防ぐ手を打つ
 - それ以外の場合
 - ⇒ 評価値の高いマスに打つ

88

宿題:3目並べの着手選択

<table border="1" style="border-collapse: collapse; width: 30px; height: 30px;"> <tr><td>○</td><td></td><td>×</td></tr> <tr><td>○</td><td>×</td><td></td></tr> <tr><td>○</td><td></td><td></td></tr> </table>	○		×	○	×		○			<table border="1" style="border-collapse: collapse; width: 30px; height: 30px;"> <tr><td>○</td><td>×</td><td></td></tr> <tr><td></td><td>×</td><td></td></tr> <tr><td></td><td>○</td><td>○</td></tr> </table>	○	×			×			○	○	<table border="1" style="border-collapse: collapse; width: 30px; height: 30px;"> <tr><td>○</td><td>×</td><td></td></tr> <tr><td>×</td><td>○</td><td>○</td></tr> <tr><td></td><td></td><td></td></tr> </table>	○	×		×	○	○			
○		×																											
○	×																												
○																													
○	×																												
	×																												
	○	○																											
○	×																												
×	○	○																											
自分にリーチ	相手にリーチ	それ以外																											

+2	+1	+2
+1	+4	+1
+2	+1	+2

得点表の例

89