

第 1 回 GAT2016 ポスター発表会

人狼知能

- ・多視点から見た心理モデルを持つ人狼知能エージェント
中村徳亜
 - ・複数のルールによる評価を総合的に判断する人狼エージェントの実装
箕輪峻，狩野芳伸
 - ・人狼 AI 環境におけるセオリーの検証
神田直樹，伊藤毅志
-

ミニ四駆 AI

- ・加速度センサを用いた坂道検出による自動制御
佐土瀬寛
 - ・ミニ四駆のためのモーター駆動制御
神津隆大
 - ・手動編集によるミニ四駆 AI
原田貴史
 - ・加速度センサーによる速度制御を施したミニ四駆 AI
折原レオナルド賢
 - ・逆転回路ミニ四駆
GHITA ATHALINA
 - ・PSoC 5 LP Prototyping Kit を用いた AI ミニ四駆の制御
里見 優太，野中 智矢
-

ターンベース戦略ゲーム AI

- ・ターン制戦略ゲーム探索のこのへんファジィ評価を用いた効率化
武藤 孝輔, 西野 順二
-

デジタルカーリング

- ・深層学習を用いたデジタルカーリングプログラム Deep Imitation System の提案
益子 直, 成澤 和志, 篠原 歩
 - ・デジタルカーリングにおける戦略的知識を用いた候補手の絞り込み
城戸皓己, 伊藤毅志
 - ・デジタルカーリング AI における戦略書に基づく序盤定石の構築
森健太郎, 伊藤毅志
 - ・デジタルカーリング AI 「じりつくん」
加藤 修, 飯塚 博幸, 山本 雅人
-

5 五将棋

- ・思考ゲーム AI 制作を行う授業の取り組み
金井誠, 岡田 拓也, 樋口 利幸, 宮下 優希, 伊藤 毅志
-

多視点から見た心理モデルを持つ 人狼知能エージェント

中村 徳亜
広島市立大学

本エージェントは、自分の考えモデル化した自己モデルと他者の考えの予測をモデル化した他者モデルを用いて行動を決定する。この2つのモデルを利用することで、自分からだけでなく他者から自分を含む他者がどのように見えているかを考え行動することができる。

Construction of AI wolf which have psychological model considering from a variety of perspectives

Noritsugu Nakamura
Hiroshima City University

My agent decides the behavior using the self-model that simulates thoughts of its own and the other-model that simulates thoughts of others. By using the two models, it is possible to decide behavior by considering other players' point of view.

1. エージェント概要

本エージェントは、自分が他者をどの役職だと思っているかをモデル化した、「自己から見た他者の心理モデル(自己モデル)」と

他者が他者(自分を含む)他者をどの役職だと思っているかをモデル化した「他者から見た他者の心理モデル(他者モデル)」の2つの心理モデルを用いて行動を決定する。

2. 多視点から見た心理モデル

2.1 心理モデルの概要

本エージェントで用いる心理モデルは、あるプレイヤーから見て他のプレイヤーをどの役職だと考えているかを確率で表現する心理テーブルから構成される。心理テーブルでは、特定の状況における発話や行動に、その発話や行動を行ったプレイヤーがどの役職であることを示唆するもののなかを意味する点数がつけられており、その点数により確率を計算する。例えば、あるプレイヤーが占い師の役職表明を行った場合、そのプレイヤーは占い師、狂人、人狼の3つの役職の可能性が高く、村人、霊能者の可能性が非常に低くなる。そのため、占い師、狂人、人狼に高い点数を付け、村人、霊能者に低い点数を付けている。そして、任意のプレイヤーによって対象となる発話や行動が行われるごとに、関連するプレイヤーの指定された役職に設定された点数が加算されていく。本モデルでは、プレイヤーごとに全役職の合計点で各役職の点数を正規化したものを、役職を推定する際の確率とみなす。なお、役職 r の点数の初期値は以下の式で定義される。

$$\text{役職 } r \text{ の初期値} = \frac{n_r}{N-1} \times 100 \quad (2.1)$$

ここで n_r は自分を除いたときの自分と同じ役職 r のプレイヤーの人数であり、 N は全プレイヤーの合計人数である。ただし、狩人は自ら狩人であると役職表明することが少なく、特定の発話・行動がほとんど見られないため村人に含める。心理テーブルの例を表1に示す。括弧の中の数字は点数を表している。

本モデルでは、自己から見た他者の心理テーブルを持つ自己モデルと、他者から見た他者の心理テーブルを持つ他者モデルが存在する。その心理モデルのイメージを図1に示す。以下で、それぞれについて述べる。

表1 心理テーブルの例

	村人	占い	霊能	狂人	人狼
A	0.00 (0)	0.80 (80)	0.00 (0)	0.10 (10)	0.10 (10)
B	0.70 (140)	0.00 (0)	0.00 (0)	0.10 (20)	0.20 (40)
...	...	...	...	...	...
N	0.00 (0)	0.00 (0)	0.00 (0)	0.40 (120)	0.60 (180)

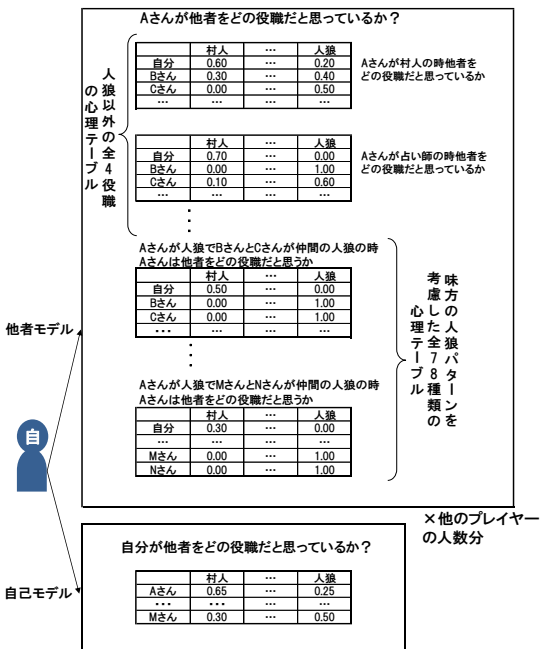


図1 心理モデルのイメージ(自分:村人)

2.2 自己から見た他者の心理モデル

自己から見た他者の心理モデル(自己モデル)は、自分が他のプレイヤーをどの役職だと思うかをモデル化したものであり、基本的に1つの心理テーブルを持つ。自分が、占い師、霊能者、狂人のような、人狼ゲームの設定上1人しか存在しない役職の場合、自己モデルにおける心理テーブルのその役職の点数は0となる。また、自分が人狼の場合、人狼であるプレイヤーを知ることができるため、味方の人狼であるプレイヤーに対してはそのプレイヤーに対する心理テーブルの人狼

の点数を 100, その他の役職の点数を 0 と設定する. さらに, 他のプレイヤーの人狼の点数は 0 と設定される.

基本的に自己モデルは自分 1 人に対して 1 つの心理テーブルしか持たないが, 狂人の場合は他の役職を騙るために, 自身が狂人の場合の心理テーブルに加え, 自分を村人としたときの心理テーブル, 自分を占い師としたときの心理テーブル, 自分を霊能者としたときの心理テーブルの 3 つを持つ.

また, 自分が人狼の場合, 人狼の心理テーブルに加え, 自分を村人としたときの心理テーブルを持つ. この心理テーブルでは, 味方の人狼を考慮しない.

2.3 他者から見た他者の心理モデル

他者から見た他者の心理モデル(他者モデル)は, 自分以外の各プレイヤーが自分を含めた他者をどの役職だと思っているかをモデル化したものであり, 複数の心理テーブルを持つ. ゲーム序盤では, 他者がどの役職であるかが不明であるため, 自分以外の各プレイヤーが村人, 占い師, 霊能者, 狂人, 人狼の 5 つ役職であると仮定した心理テーブルを作成する. なお, 自分が占い師, 霊能者, 狂人の場合, 自分以外のプレイヤーにその役職が存在しないため, その役職と仮定した心理テーブルは不要である. また, 他者を人狼と仮定したとき, その味方の 2 人の人狼も仮定する必要があるため, 人狼 3 人の全てのパターンの心理テーブルを作成する. この人狼のパターンに対応する心理テーブルは各プレイヤーを視点とし, 自分と視点となるプレイヤー以外の 13 人のプレイヤーの内 2 人を選択する全てのパターンの心理テーブルを作成するため, ${}_{13}C_2 = 78$ 通りとなる. この人狼の心理テーブルはゲームが進行していく中で不要となったものは削除されていく. 削除される条件は,

1. 全ての人狼プレイヤーが既に死亡している場合
2. 人狼だと確定したプレイヤーが含まれ

ていない場合

3. 人間だと確定したプレイヤーが含まれている場合

である. 1 は, 心理テーブルにおける人狼のプレイヤーが全て死亡しているとき, ゲームが終了していないということはその人狼のパターンの中に少なくとも 1 人の人間が含まれていたことになるため, このパターンに対応する心理テーブルは削除される. 2 は, 自分の特殊能力結果で人狼と判明した時など人狼だと確定したプレイヤーを含んでいない時, その人狼のパターンは間違いであるため, このパターンに対応する心理テーブルは削除できる. 3 は, 襲撃により喰われたプレイヤーや自分の特殊能力結果で人間と判明した時など人間だと確定したプレイヤーが人狼パターンに含まれている時, その人狼のパターンは間違いであるため, このパターンに対応する心理テーブルは削除できる.

2.4 信頼度の算出

心理テーブルに点数を加算する際, 発話や行動を行ったプレイヤーや対象となったプレイヤーがどれくらい信頼できるかによってその発話や行動の心理テーブルへの影響が変化する. 例えば, とても信頼している占い師が人狼だと占ったプレイヤーとあまり信頼していない占い師が人狼だと占ったプレイヤーでは前者に占われたプレイヤーの方がより疑わしいと考えられる. そのため, 心理テーブルに点数を加算する際, 各プレイヤーをどの程度村人陣営, または人狼陣営だと考えているのかを表す信頼度を算出する必要がある. 信頼度は以下の式により計算する.

$$\text{信頼度} = (\text{村人の確率} + \text{占い師の確率} + \text{霊能者の確率}) - (\text{狂人の確率} + \text{人狼の確率}) \quad (2.2)$$

村人陣営であることが確定したときは, 信頼度が最大値をとり 1.0 となる. また,

人狼陣営であることが確定したときは、信頼度が最小値をとり-1.0 となる。この信頼度は、心理テーブルが更新されるたびに更新し、心理テーブルへの加点や、行動決定に使用する。

2.5 他者モデルにおける心理テーブルの確信度

2.5.1 他者を人狼以外だと仮定した時の他者モデルにおける心理テーブルの確信度

他者モデルは、他の各プレイヤーが村人、占い師、霊能者、狂人、人狼すべての役職の可能性を考慮するため、心理テーブルが複数存在する。しかし、複数の心理テーブルの中で最も重要な心理テーブルはそのプレイヤーの本当の役職と一致するものである。また、他者モデルの心理テーブルから確率を参照する際、その確率がどの程度信頼できるものであるかを判断する必要がある。例えば、占い師だと表明しているプレイヤーの他者モデルを参照する際、そのプレイヤーが村人、霊能者である可能性は非常に低いため、そのプレイヤーを村人、霊能者と仮定した時の心理テーブルの確率はあまり信頼できないと考えられる。そのため、他者モデルにおける心理テーブルの確信度が必要となる。あるプレイヤーを村人、占い師、霊能者、狂人と仮定した時の心理テーブルの確信度は、自己モデルから、他者モデルの視点となるプレイヤーの仮定した役職の確率を確信度として使用する。

2.5.2 他者を人狼だと仮定した時の他者モデルにおける心理テーブルの確信度

他者を人狼だと仮定したとき、人狼 3 人のパターンの全ての心理テーブルを作成しているため、その全てのパターンの心理テーブルに確信度が必要となる。そこで、人狼のパターンの心理テーブルにも別の点数付けを行い、人狼のパターンに対応する心理テーブルの確信度を算出することとする。

ここでの確信度は、人狼がそのパターンであるということがどれだけ確からしいかということの意味する。人狼のパターンに対応する心理テーブルの確信度の初期点は 50 点とし、発話や行動により発話(行動)者と発話(行動)対象者のどちらもが人狼である度合いを 0~100 点の間で点数とし任意のパターンの心理テーブルに加点していく。この点数を、人狼のパターンの心理テーブルの合計点で正規化した確率を人狼パターンに対応する心理テーブルの確信度とする。

例えば、AさんとBさんのどちらもが占い師だと役職表明をしている場合、AさんとBさんの両方が人狼である度合いは低いと考えられるため、低い点数を設定する。加点をする際、加点する点数が 50 以上の時、AさんとBさんのどちらも含む人狼のパターンの心理テーブルの確信度に点数に加点する。50 点より小さい時、「50+加点する点数」をAさんとBさんのどちらかを含まないかどちらも含まない人狼のパターンの心理テーブルの確信度に加点する。この確信度を用いることで、Aさんが人狼である時、Bさんも人狼であるといったような組み合わせの確率をとることができないという自己モデルの弱点を補うことが可能となる。

2.6 心理テーブルの数値変更

心理テーブルの確率計算のための数値は、発話、投票、特殊能力の結果から加点をする。発話では、役職表明、占い結果、霊能結果、役職予想を、特殊能力は、自分の占い結果、霊能結果を使用する。発話や行動が行われるたびに、対応する点数を心理テーブルに加点していく。このとき、加点する点数が 0 点の時その役職ではないと考え 0 を代入する。また、既に心理テーブルの値が 0 であるときその役職ではないということになるのでその役職への加点は行わない。

また、会話が進行していく中で、役職を騙る人狼や狂人の発話に矛盾が生じることが

ある。矛盾が生じる状況の例を以下に示す。

1. 人狼だと占っていたプレイヤーが人狼に襲撃された占い師表明プレイヤー
2. 自分が人狼でないとき、自分のことを人狼だと占った占い師表明プレイヤー
3. ある占い師だと表明しているプレイヤーを占い師と仮定したとき、そのプレイヤー以外の占い師だと表明しているプレイヤー、そのプレイヤーの能力結果と矛盾した能力結果を発言したプレイヤー、仮定した占い師が人狼だと言ったプレイヤーの人数が人狼陣営の人数である 4 人を超えた場合の占い師表明プレイヤー
4. 自分が特殊能力を持った役職の時、自分の能力結果と矛盾した能力結果の発言をした役職表明プレイヤー

これらのプレイヤーは、明らかに矛盾した発言をしており人狼陣営であることがわかる。そのためそのプレイヤーの村人、占い師、霊能者の点数を 0 点にする。また、占い師、霊能者、狂人のような 1 人しか存在しない役職が確定した時、心理テーブルのそのプレイヤー以外の確定した役職の点数を 0 にする。

3 エージェントの行動選択

3.1 役職予想

本エージェントは、自己モデルと他者モデルを使用して役職予想を行う。占い師、霊能者だと表明しているプレイヤーが存在する時、最も占い師だと思うプレイヤー、最も霊能者だと思うプレイヤーを自己モデルと他者モデルから決定する。占い師だと表明したプレイヤーを例に説明する。まず、占い師だと表明したプレイヤー以外の生きているプレイヤーの他者モデルの心理テーブルから、その表明したプレイヤーが占い師である確率と、その他者モデルにおける心理テーブルの確信度の積を求める。この計算

をその表明したプレイヤー以外の全ての生きているプレイヤーの他者モデルの全ての心理テーブルで行い、それらを平均した値を占い師だと表明しているプレイヤーの占い師スコアとする。全ての占い師だと表明しているプレイヤーで占い師スコアを算出し、最大の値であるプレイヤーを占い師だと考える。霊能者についても同じ計算をして最も信頼できる霊能者を選ぶ。このとき選んだ占い師と霊能者をここでは、仮占い師、または仮霊能者と呼ぶ。また、仮占い師、仮霊能者以外の生きているプレイヤーの中から、同様の方法で最も人狼だと思うプレイヤー(仮人狼)を選ぶ。

3.2 投票

3.2.1 村人陣営

投票者の選択は、他者モデルにおける人狼パターンの心理テーブルを使用し決定する。役職表明をしているプレイヤーが 2 人以上存在する場合、仮占い師と仮霊能者以外のプレイヤーは狂人か人狼であると考えられる。狂人は自分の死亡が人狼陣営の敗北に直接影響を与えないため、偽の役職表明をする可能性が高いと考えられる。そのため、仮占い師と仮霊能者以外の役職表明したプレイヤーに狂人が含まれる可能性は高いと考えられる。そこで、3.1 節の役職予想と同様の方法で、役職表明したプレイヤーの中から最も狂人だと考えられるプレイヤー(仮狂人)を選択する。これにより、役職表明をしているプレイヤーの中で、仮占い師、仮霊能者、仮狂人以外のプレイヤーは人狼である可能性が非常に高くなる。そこで、他者モデルから、人狼の可能性が高いプレイヤーを含む人狼パターンの心理テーブルの中で心理テーブルの確信度が最も高いものを選ぶ。その心理テーブルの人狼であるプレイヤーのうち、生きているプレイヤーを選択し投票する。もし、役職表明しているプレイヤーが 3 人以上存在せず、人狼の可

能性が高いプレイヤーを選択することができなかった場合、3.1 節の役職予想により選択した仮人狼を使用し、人狼パターンの心理テーブルを選択する。

3.2.2 狂人

自分が狂人の場合、自分が騙っている役職と同じ役職の表明を行っているプレイヤー、自分の特殊能力の結果と矛盾した能力結果の発話をしているプレイヤーに優先して投票する。該当するプレイヤーが存在しないとき、最も確信度の高い人狼パターンの心理テーブルを取得し、その心理テーブルにおいて人狼であるプレイヤー以外のプレイヤーから最も人狼の確率が高いプレイヤーに投票する。

3.2.3 人狼

自分が人狼である時、自分が村人だとしたときの心理テーブルから、信頼度が-1.0、すなわち、自分と矛盾しているプレイヤーに優先して投票する。そのプレイヤーが存在しない場合、人狼以外のプレイヤーから、人狼の確率が高いプレイヤーを 3 人選択する。自分がその 3 人のそれぞれに投票したと仮定した際に他者から自分がどのように見えるかを考えて投票先を決定する。具体的には、生きている全てのプレイヤーの他者モデルの全ての心理テーブルから自分がその 3 人のそれぞれに投票した際の自分に対する信頼度とその心理テーブルの確信度の積を算出する。それらを 3 人の対象者ごとに平均し、その平均が最も高くなるプレイヤーを選択し投票を行う。すなわち、他者から見た自分の信頼度が高くなるように投票するプレイヤーを選択する。

3.3 役職表明

自分が占い師、霊能者の場合の役職表明のタイミングは

- 誰も役職表明していない場合

- 他のプレイヤーが自分の役職の表明をした場合
- 自分の特殊能力の結果で人狼を見つけた場合

であり、この 3 つの状況の時、役職表明を行うかどうかの確率を設定しておき、その確率に従って役職表明を行うかどうかを決定する。

自分が狂人の場合、周りの役職表明しているプレイヤーの人数ごとに占い師だと役職表明する、霊能者だと役職表明する、何も役職表明しないに確率を設定し、その確率をもとに役職表明を行うかどうかを決定する。

自分が人狼の場合、他の占い師だと役職表明しているプレイヤーに人狼だと占われた場合のみ狩人だと役職表明、または、村人だと役職表明をする。

3.4 特殊能力

3.4.1 占い師

自分が占い師の場合、占い先の選択は、3.2 節の投票で使用した人狼パターンの心理テーブルにおいて人狼であるプレイヤーの中から、投票しようと考えているプレイヤーの他に、まだ占っておらず、生きているプレイヤーが存在する場合そのプレイヤーを占う。もし存在しないとき、仮人狼を占う。それでも該当するプレイヤーが存在しない場合ランダムに選択する。

3.4.2 狩人

自分が狩人の場合、護衛先の選択は、仮占い師が生きている場合、そのプレイヤーを優先的に護衛する。死亡している場合、仮霊能者が生きているならばそのプレイヤーを護衛する。どちらも死亡している場合、占い師から人間だと占われたプレイヤーを優先して護衛する。該当するプレイヤーが存在しない場合、6.2 節の投票で使用した人狼パ

ターンに含まれないプレイヤーの中からランダムで選択する。

3.4.3 狂人

自分が狂人の場合、自己モデルと他者モデルから人狼だと仕立て上げるプレイヤーを3人選択する。この3人は、自分と同じ役職を騙っているプレイヤー、自分と矛盾した能力の発話を行っているプレイヤー、自分が人狼だと既に占っているプレイヤーを考慮して選択する。この3人の結果を人狼、そのほかのプレイヤーの結果を人間であるとする。この時、偽の占い対象の選択は、まだ占っておらず生きているプレイヤーの中から3.1節の役職表明で述べた方法で人狼の可能性が高いプレイヤーを選択する。霊能結果の発話の場合、霊能対象が先ほど選択した3人の場合人狼、そのほかの場合、人間であると言う。

3.5 人狼の襲撃先

襲撃するプレイヤーの選択は、自分を村人だと仮定した時の心理テーブルから信頼度の最も高いプレイヤーを優先して襲撃する。ただし、役職表明しているプレイヤーは占い師、霊能者でそれぞれ1人ずつまでしか襲撃しない。

複数のルールによる評価を総合的に判断する 人狼エージェントの実装

箕輪峻¹ 狩野芳伸¹

¹ 静岡大学情報学部

概要

人狼ゲームでは単一の情報だけでプレイヤーの陣営を予想するのは難しい。本エージェントでは複数のルールによる評価を元に総合的に判断する評価システムの構築を行った。

はじめに

現在、チェスや将棋をはじめとした様々な問題に対して人工知能が人間と同等かそれ以上の成績を収めている。そこで篠田らは人狼を人工知能にプレイさせる人狼知能プロジェクトを立ち上げ、汎用人工知能の新たな標準問題として人狼ゲームを提案した[篠田 14]。人狼知能プロジェクトでは過去に CEDEC2015 にて第 1 回人狼知能大会が開催されたほか CEDEC2016 にて第 2 回人狼知能大会が予定されている。以下では狼知能ミニ大会に対してチーム名「kanolab」で登録したエージェントの設計について解説する。人狼知能プロジェクトにおける人狼ゲームは人間が行う人狼ゲームと比較して発言内容が限定されている。そのため各陣営のエージェントが取れる戦略は実際の人狼ゲームと異なり他エージェントとの協調を前提としない単純なものに限られる。単純な戦略のみであればルールベースで記述可能と考えられることから、本エージェントでは完全ルールベースによる実装を行った。

手法

本エージェントは「評価」と「行動」の二つから構成されている。評価は全役職共通のもので、複数の評価ルールに基づき各エージェントの人狼陣営らしさを数値化する。行動では評価によって得られた評価値等を元に投票先や各役職の能力対象の選択を行う。なお、本エージェントは以下の行動を想定しない。

- 村騙り
- 騙り CO 出現時の真潜伏
- スライド CO

これらは実際の人狼ゲームにおいても推奨されない行動であり、エージェント間の意思疎通が困難な本人狼ゲームにおいて発生する可能性が極めて低いことから除外して考える。

評価

評価では各エージェントについて人狼陣営である可能性が高いほど値が大きくなるように評価値を算出する。最終的な評価値はエージェント自身の行動を評価する単独評価値とエージェント間の関係性を評価するライン評価値の二種類を組み合わせて得る。あるエージェント n の単独評価値 P_n は i 種類の評価ルールが存在するとき、各評価値 X に重み a をかけた値の合計で表される。

$$P_n = \sum_i (X_{i_n} \cdot a_i)$$

ライン評価値も同様に複数のルールから得られた評価値に重みをかけたものの合計で求め、最終的な評価値はこの二つを元に得る。最終的な評価値を各評価ルールの評価値に重み付けしたものの総和で表すことで、新たな評価ルールの追加や各評価ルールの重要度の調整を容易に行うことを可能にした。

行動

各役職の行動は前述の評価値等を元に決定される。以下では各役職の基本的な動きを解説する。

● 村人

村人の行える行動は発話と投票のみである。投票は自身を除く生存している全エージェントの中で評価値の最も高いエージェントに対して行う。投票先の決定は他の村人陣営の役職でも共通となる。発話は投票先の宣言のほか、評価値が最も低いプレイヤーを村人と予想する、複数の CO が存在する役職について評価値の低い方を真と予想するなどの発話を行う。

● 狩人

狩人の行動は護衛先の決定以外は村人と共通となる。また、人狼陣営に狩人の生存状態を知られないために、騙り CO が出た場合以外は投票先に指定される状況でも CO は行わない。護衛対象の優先度は①占い師、②霊能者、③その他となり

各段階において最も評価値の低いエージェントを護衛する。ただし、①, ②について複数の CO が存在し、評価値が最も低いエージェントが死亡している場合は残りのエージェントは候補から除外し、残りの候補から護衛先を決定する。

- 占い師

占い師は初日の最初の発話で必ず CO する。占い先には CO しておらず自分がまだ占っていないエージェントの中で最も評価値の高いエージェントを選択する。上記の条件で候補者が存在しない場合、霊能者 CO しているエージェントを候補者に含める。占い師 CO したエージェント以外の全エージェントを既に占っている場合は、対抗 CO しているエージェントを占い先として指定する。占い師は CO 後、まだ発話していない占い結果を順番に全て発話する。また、占い結果が白だったエージェントの評価値が全エージェントの中で最も高かった場合、狂人と予想する。

- 霊能者

霊能者は基本的には最初の処刑が行われた翌日に CO する。ただし、自身が処刑対象となる可能性が高い場合や騙り CO が出現した場合は初日でも CO する。占い師と同様に、CO 後は発話していない結果が存在する場合は優先的にそれを発話する。また、自身の結果と占い師の占い結果が矛盾している場合はその占い師を偽物と発言する。

- 狂人

誰かが占い師 CO してから次の自分の発話タイミングまでに対抗 CO が出していない場合占い師 CO する。すでに占い師 CO したエージェントが複数存在する場合は霊能者 CO する。占い師 CO 後は占い師と同じ選択方法で選択した対象にランダムで判定を出す。霊能者 CO した場合も同様に確率で黒判定を出す。投票は確定黒のエージェントを除いたエージェントから評価値をもとに確率的に投票する。

- 人狼

CO しない。人狼陣営を除いて最も評価値の高いエージェントに投票する。ただし、確定黒を出された人狼陣営のエージェントが存在する場合はそのエージェントに投票する。襲撃対象は評価値の最も低いエージェントを対象とする。役職 CO しているエージェントが存在する場合は優先的に襲撃対象とするが、確定で襲撃した場合狩人に防御される可能性があるため残りの人数に応じて確率的に襲撃する。生存エージェントが 7 人か 5 人の場合のみ翌日の生存人数を偶数人にするために襲撃対象を最も投票が集まりそうなエ

ージェントを指定する。

まとめ

本エージェントでは複数のルールを利用し総合的に評価を行うことで様々な情報を加味した行動選択が可能にした。しかし、各ルールに設定した重みの値は経験的に決定したものでありその妥当性は不明である。したがって、今後十分なサンプルをもとに適切な学習を行うことで精度の向上が期待できるとともに、どのような情報が陣営の特定に必要なかを明らかにできる可能性がある。

参考文献

[篠田 14] 篠田孝祐・鳥海不二夫・稲葉通将・大澤博隆・片上大輔. (2014). FAN-14-016 人工知能標準問題としての人狼ゲームの提案(OS: 人狼知能研究). インテリジェントシステム・シンポジウム講演論文集, 2014(24), 74-77. 一般社団法人日本機械学会.

人狼 AI 環境におけるセオリーの検証

神田直樹 伊藤毅志

電気通信大学

概要：人狼ゲームでは“セオリー”と呼ばれる戦略の指針が存在するが、人狼 AI 環境でどのセオリーが有効であるか不明である。そこで本研究では、基本的な行動を行う“基準プレイ”を提案し、基準プレイとセオリーの勝率を比較する事で、セオリーの検証を試みた。

Verification of Accepted Theories in Werewolf AI Environment

Naoki Kanda Takeshi Ito

The University of Electro-Communications

Abstract: In playing werewolf, a strategic policy called "theory" exists. but it is unknown which a theory is effective in werewolf AI. In this research, we proposed a "basic play" that plays basic behavior. We try to verify the validity of the theories by making comparison between the basic play and the play by theories.

1. はじめに

「人狼ゲーム」とは村人陣営とそれに紛れ込んだ人狼陣営とが、相手の正体を探りあいながら自陣営の勝利を目指す、不完全情報型のコミュニケーションゲームである。近年では学術分野でも注目を集めており、例えば人狼を多角的な視点から研究を行う「人狼プロジェクト」が立ち上がっている[1]。このプロジェクトでは、人狼ゲームを人工知能に行わせるために擬似会話プロトコルを備えた人狼 AI の対戦環境として人狼知能プラットフォームを提供しており、このプラットフォームを用いた大会も実施されている[2]。

人狼ゲームではゲーム進行に応じたプレイの指針を「セオリー」と呼び、陣営や役職ごとの「セオリー」が模索されている。人間同士の人狼ゲームにおいては、このセオリーに従ってプレイすることで勝率が上がると考えられている。人狼 AI においても、セオリーを導入する事が有効であると推測される。しかし、セオリーには膨大な数が存在し、人狼 AI を強くするためにどのようなセオリーを導入するべきなのか不明である。

そこで本研究では、基本的な役職行動と投票行動を行う“基準プレイ”を提案し、人狼知能プロジェクトが提供する人狼知能プラットフォームで用意されている自動対戦可能なシミュレーション機能を用いて、人狼ゲームにおいてセオリーとされているプレイをこの基準プレイに付与することで、実際に勝率が上がるのかを調べ、そのセオリーの優劣を検証していく。

2. 人狼ゲーム

2.1 ルール

ここでは人狼プロジェクトが提唱している人狼標準ルールに沿って説明をする[3]。プレイヤーは開始時に人狼陣営（人狼 3、狂人 1）か村人陣営（村人 8、占い師 1、霊媒師 1、狩人 1）に振り分けられ、ゲームは図 1 のように昼夜の 2 フェーズで構成される。

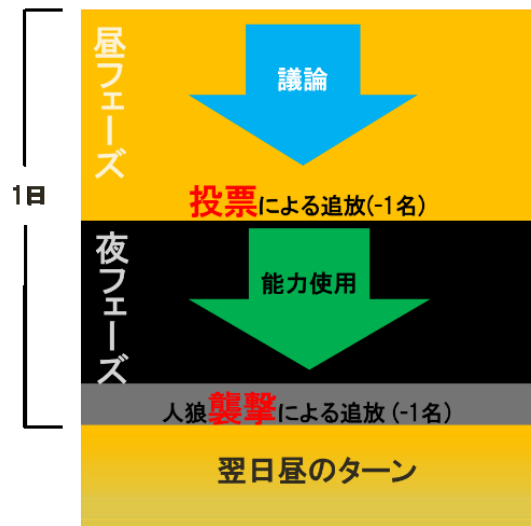


図 1 人狼ゲームの流れ

昼フェーズではプレイヤー同士の話し合いによって追放者を決める「投票」が行われ、夜フェーズでは、特殊能力を持つ役職による特殊能力、人狼の「襲撃」による追放が行われる。追放されたプレイヤーはその後ゲームに関与することは出来ない。勝利条件は、人狼陣営は

人狼の残人数が人間陣営の残り人数を上回ることで、人間陣営は人狼の残り人数を0にすることである。

人狼ゲームでは役職ごとに、獲得できる情報や可能な行動が異なる。「占い師」は夜フェーズに指定したプレイヤーの人狼か人間かを知ることが出来る。「霊媒師」は追放されたプレイヤーの人狼か人間かを知ることが出来る。また、「狂人」は能力を持たず、判定結果は村人陣営となるが、人狼陣営の勝利が自身の勝利となる。

2.2 人狼ゲームにおけるセオリー

人狼ゲームにおける各陣営に最善と思われる一連の流れや戦略を人狼用語でセオリーと呼んでいる。人狼ゲームにおける陣営・役職共通のセオリーは、大きく分けて以下2つの行動をゲーム進行に応じて決定する行動指針である。

- 自分が何の役職であるかを場に情報としてカムフラグアウト(以下 CO)する
- 役職ごとに与えられている特殊能力の対象プレイヤーをどのプレイヤーにするか

村人陣営の目的は場に残っている人狼を昼の投票で追放する事である。そのため、村人陣営の役職は場に残っているプレイヤーの情報を明らかにしていくことが必要である。占い師の基本的なセオリーとして、ゲームを開始してすぐにCOを行い、自分が一度も選んでいないプレイヤーを占い対象とすることが考えられる。これは、占い師が場のプレイヤー1人の人狼かどうかの情報を明らかに出来る唯一の役職であり、場に多くのプレイヤーの情報を残すことが村人陣営に有利に働くからである。また、霊媒師の基本的なセオリーとしては、追放されたプレイヤーが人狼だった時にCOを行う事が考えられる。これは、昼に投票によって追放されたプレイヤーの人狼かどうかを知ることが出来る役職であり、人狼が追放されたという情報は村人陣営に有益な情報であり、むやみにCOを行うと人狼に追放される恐れがあるためである。

人狼陣営の目的は人狼が出来るだけ多くまで生き残ることである。そのため、人狼陣営の役職は村人陣営の役職の特殊能力による情報の信用を下げ、役職を騙ることで場に多くの偽の情報を残す事が求められる。狂人の基本的なセオリーとしては、占い師としてゲーム開始時にCOを行い、ランダムに村人判定を出す事が考えられる。これは、占い師が複数存在する状況を作り、本物の占い師の情報の信用をより下げることが人狼陣営にとって有利に働くからである。人狼の基本的なセオリーとしては、COを行わずに村人として振る舞い、村人陣営に有益な存在を襲撃する事が考えられる。これは、村人陣営にとって有益な存在を減らすことは人狼陣営にとって有利に働くからである。

しかし、これらの基本的なセオリーは一般的に考えられるセオリーの1つであり、陣営毎の目的に応じて様々なセオリーが存在する。例えば、占い師の判定の信用をより下げるために、人狼の一人が初日に占い師としてCOするセオリーがある。村人陣営にもCOを行う前に追放される事を防ぐために霊媒師が初日にCOを行うセオリーなどがある。

このように人狼ゲームには多種多様なセオリーがあり、各プレイヤーはその場に応じた適切なセオリーを選択する必要がある。

3. 関連研究

コミュニケーションゲームにおいて、セオリーを検討する研究が行われている。例えば、Bravemanらは人狼に似たゲーム“Mafia”を用いて、役職が3種類の場合における各役職の最適戦略を確率論によって求めた[4]。西野はこの研究を人狼ゲームに適用し、加えて、“人狼同士はお互いに投票しない”というセオリーを追加する事で、人狼陣営の勝率がさらに高くなる事を示した[5]。

しかしながら、これらの研究では議論を単純化するために検討する役職を村人・人狼のみに限定しており、人狼標準ルールなどの一般的な人狼ゲームにそのまま適用することは出来ない。そのため、最適なセオリーをこれらの研究のような手法で求める事は難しい。そこで本研究では、セオリーを導入したAIによる対戦を繰り返し行わせ、勝率を求めることで、最適なセオリーを求める方法を採用する。

4. 基準プレイ

セオリーを検証する際に基準となるプレイを行う基準プレイを定義した。基準プレイは、昼フェーズに誰に投票するかを決定する投票行動・役職ごとに与えられている特殊能力の対象プレイヤーを決定する役職行動・自分が何の役職であるかをどのタイミングで宣言するか決定するCOタイミングの3つの行動で構成されている。役職行動・COタイミングは合理的な行動をとるように設定されている。また、投票行動は、占い師・霊媒師の発言によって得られた情報に応じた他プレイヤーの人狼らしさの度合い、疑い度に従って行動するように設定されている。

4.1 疑い度

疑い度は、占い結果などの公開された真偽不明の情報によって変動する、各プレイヤーの人狼らしさの度合いである。

本研究の基準プレイでは、占い師・霊媒師による特殊能力による情報が無い状態の疑い度の初期値は灰色とする。占い師の判定結果は場のプレイヤーの疑い度を変更できる情報とする。占い師の判定結果が人間の場合は対象プレイヤーの疑い度が1段階白に近づくものとする。人狼判定の場合は対象プレイヤーの疑い度が1段階黒に近づくものとする。この疑い度は、場にCOした占い師の人数分の段階で表される。図2は、占い師が2人COした場合の疑い度を示している。二人共同対象プレイヤーに人間判定を出した場合は2段階白に近づくことで確定人間となり、人狼判定を出した場合は2段階黒に近づくことで確定人狼となる。このように、占い師が2人COした場合全てのプレイヤーは5段階の疑い度で表される。占い師が3人COした場合は、3人の人間判定により確定人間となるため、全てのプレイヤーは疑い度を7段階で表わされる。



図2 占い師2名COした場合の各プレイヤーの疑い度

霊媒師の判定結果は占い師が本物かどうかを判断するために使用される。前日の昼に追放されたプレイヤーを対象とした占い師の判定結果と霊媒師の判定結果が異なる場合、占い師と霊媒師の判定結果に矛盾が生じるため、占い師は偽物と考えられる。偽物の占い師が出している判定結果は一切疑い度に影響を与えないものとし、偽物の占い師は一段階疑い度が黒に近づく。場に霊媒師が1人しかCOしていない場合は、その霊媒師1人の判定結果と占い師の判定結果を照らしあわせて、矛盾が起きているか判別する。また、場に霊媒師が2人以上COしている場合は、霊媒師全員の判定結果が一致した時にその霊媒師の判定結果と占い師の判定結果を照らしあわせて、矛盾が起きているか判別する。具体的には、霊媒師2人がCOし、2人とも前日の昼フェーズに追放されたプレイヤーを対象に人狼判定の結果を出した場合、霊媒師全員の判定結果が一致したとみなされ、追放プレイヤーに人間判定を出した占い師は結果の矛盾により偽物の占い師になる。偽物を除いて、場に残った占い師が1人になった場合には、その占い師は”確定占い師”となり、確定占い師の疑い度は白になる。

4.2 投票行動

本研究で定義する基準プレイでは、他のプレイヤーへの疑い度によって投票先を決定する。各プレイヤーは図3のように、疑い度が高いほど投票優先順位が高いと見なし、順位が最も高いプレイヤーに投票を行うものとする。同じ投票優先順位にいるプレイヤーが複数存在している場合、その中からランダムで投票先を決める。



図3 疑い度による投票先の決定

また、人狼プレイヤーも基本的にこの投票行動に従うが、同じ優先順位のプレイヤーが複数人居た場合は、そ

の中から人狼でないプレイヤーへ投票を行う。

4.3 役職行動

本研究で定義する基準プレイにおける各役職の行動は、疑い度に基づいて以下のように行動する。

【占い師】

初日にCOし、他占い師と自分が一度占ったプレイヤー以外を対象にランダムで占う。

【霊媒師】

前日の昼フェーズの追放者が人狼であった場合、COして判定結果を公開し、それ以降は判定結果を公開し続ける。

【狩人】

特殊能力による護衛先を次の方法で選択する。

・確定占い師が存在しない場合

「占い師」「霊媒師」の中から、ランダムに護衛する。存在しない場合、最も疑い度が白に近い対象の中からランダムに護衛する。

・確定占い師が存在している場合

「確定占い師」を護衛する。対象が追放されていた場合、確定占い師が存在しない場合と同様に行動する。

【狂人】

初日に「占い師」としてCOを行う。占い先は他占い師と自分が一度占ったプレイヤー以外を対象にランダムで占う。判定結果は以下の確率で人狼判定を行う。

$$3 - \frac{\text{自分が行った人狼判定の数} - \text{霊媒師による人狼判定の数}}{\text{その日の生存人数}}$$

【人狼】

人狼は役職としてCOは行わない。人狼の特殊能力による夜フェーズの追放先は次の方法で決定する。

・確定占い師が存在しない場合

「占い師」「霊媒師」以外をランダムに襲撃する。存在しない場合は、場に生存するプレイヤーの中からランダムで襲撃する。

・確定占い師が存在する場合

以下の優先順位で、対象をランダムに襲撃先を決定する。

- ① 生存人数が7人以下の場合、「占い師」「霊媒師」「確定人間」
- ② 「霊媒師」「確定人間」
- ③ 「確定占い師」及び「人狼」以外のプレイヤー
- ④ その他の生存者

5. セオリーの検証実験

5.1 目的

本実験の目的は、人狼AI環境において、どのようなセオリーが有効なのかを検証する事である。

5.2 対戦回数

信頼できる勝率を得るためには十分な回数の対戦を行う必要がある。ここで行う対戦回数について、試合の勝敗をベルヌーイ型の事象と考え、5.1式[6]を用いて算出した。

$$n = p(1 - p) \left(\frac{S}{e} \right)^2 \quad (5.1)$$

p は事象発生確率, S は信頼区間, e は誤差である. 本実験では, 信頼できる勝率として, 信頼区間を 95% ($S = 1.96$), 誤差を 0.3% として算出した勝率と, 信頼区間を 95% ($S = 1.96$), 誤差を 1% として算出した勝率の 2 つを扱う. この時, 信頼できる勝率を求めるのに十分な対戦回数は, $p(1-p)$ の項が $p = 0.5$ で最大となることから, 誤差が 0.3% の場合は 11 万回, 誤差が 1% の場合は 1 万回となる.

5.3 方法

検証対象のセオリー通りに行動するように, 基準プレイの CO タイミング・役職行動を変更した人狼 AI を作成し, 人狼知能プラットフォームを用いて設定した対戦回数の連続対戦を行わせ, 人狼陣営の勝率を出力する. 対戦回数は 5.2 で求めた誤差 0.3% の対戦回数 11 万回を用いた. 出力された勝率を基準プレイの人狼陣営勝率と比較を行い, 基準プレイよりも有意に勝率が上がっているセオリーであるか検証を行う.

5.4 検証セオリー

本研究では人狼陣営にとって考えられるセオリー 22 個を作成・比較した. 基準プレイは人狼が騙らずに狂人が占い師として騙るセオリーとなる. これに変更を加えた以下の 5 つのセオリーの検証結果を述べる.

- I. 初日人狼が占い師として騙り, ランダムに村人判定を出すセオリー A
- II. 初日人狼 2 人が占い師として騙り, ランダムに村人判定を出すセオリー B
- III. 初日人狼が占い師として騙り, 初日に味方人狼に村人判定を出すセオリー C
- IV. 狂人が霊媒師として騙るセオリー D
- V. 狂人が霊媒師として騙り, 人狼が初日占い師として騙るセオリー E

5.5 結果

基準プレイと, 5 つのセオリーの人狼陣営勝率を表 1 に示す.

表 1 各セオリーにおける人狼陣営勝率

検証セオリー	人狼陣営勝率
基準プレイ	67.15
セオリー A	63.85
セオリー B	43.75
セオリー C	79.28
セオリー D	47.07
セオリー E	50.20

基準セオリーより 5 つのセオリーが有効であるか, 有意水準 5% での片側 t 検定を行った. その結果, セオリー C は基準プレイより有意に勝率が高く, セオリー A, B, C, D, E は, 基準プレイより有意に勝率が低かった.

5.6 考察

結果より, 人狼が騙らずに狂人が占い師として騙るセオリーと比較すると, 人狼 1 人が初日に占い師として騙り, 初日に味方人狼に人間判定を出すセオリーは人狼陣

営勝率が上がった. この原因としては, 占い師を騙った人狼が味方人狼に村人判定を出すと, 村人判定を出された人狼の疑い度が白に近づき, ゲーム序盤で人狼 1 人が投票による追放を免れ, ゲーム終盤まで生き残った事が考えられる.

この結果から, 人狼陣営の勝率を上げるための一つの方法として, 人狼の疑い度を早急に下げる必要があると考えられる.

6. おわりに

本研究では, 人狼 AI 環境を用いて人狼ゲームにおいてセオリーとされているプレイをこの基準プレイに付与することで, 実際に勝率が上がるのかを調べ, そのセオリーの優劣を検証した.

結果として, 人狼が騙らない人狼陣営セオリーと比べ, 人狼が 1 人占い師として騙り, 尚且つ味方人狼に村人判定を出すセオリーは勝率が上がり, 人狼が 1 人, または 2 人占い師として騙るのみのセオリーや, 狂人が霊媒師として行動したセオリー, 狂人が霊媒師として騙り人狼が占い師として騙るセオリーは勝率が下がった事が分かった. 今回は人狼陣営にとって考えられるセオリーの検証を行い, 人狼陣営の勝率を上げるためには人狼陣営がどのような行動をするべきか考察を行った. 同様に, 村人陣営にとって考えられるセオリーを作成し検証を行うことで, 村人陣営の勝率を上げるためには村人陣営がどのような行動をするべきか考察することが可能であると考えられる.

今後は, まだ検証を行っていない村人陣営のセオリーや, 人狼陣営の更なるセオリーなどの様々なセオリーのバリエーションを調べることで, 状況に応じてどのようなセオリーを取ることが最も良いセオリーなのかを考察していきたい.

<参考文献>

- [1] Artificial Intelligence based Werewolf, 入手先 (<http://www.aiwolf.org/>)(参照 2015-09-20)
- [2] 人狼知能コンテスト IN CEDEC2015, Artificial Intelligence based Werewolf, 入手先 (<http://www.aiwolf.org/aiwolf-cedec2015/>)(参照 2015-09-20)
- [3] 人狼知能標準ルール 20150713, Artificial Intelligence based Werewolf, 入手先 (<http://www.aiwolf.org/aiwp/wp-content/uploads/2015/07/人狼知能標準ルール20150713.pdf>)(参照 2015-09-20)
- [4] Mark Braverman, Omid Etesami, Elchanan Mossel : MAFIA: A THEORETICAL STUDY OF PLAYERS AND COALITIONS IN A PARTIAL INFORMATION ENVIRONMENT, Institute of Mathematical Statistics, 2008 Vol.18, No.3, 825-846 (2008)
- [5] 西野順二 : 自然な人狼の勝率, 情報処理学会研究報告, Vol. 2015-GI-33 No.18 (2015)
- [6] 永田端 : サンプルサイズの決め方, 朝倉書店 (2005)

加速度センサを用いた坂道検出による自動制御

佐土瀬 寛

電気通信大学 情報・通信工学科 西野研究室

概要: 加速度センサを用いることで斜体の傾きを計算することが可能となる。今大会では坂道を検出し、速度を制御するという単純な制御方法の実装を予定している。この制御により、コースアウトせず完走しやすくすることを目標とする。

1 はじめに

ミニ四駆では決められたコースをどれだけ早く走り切るかを競い合っている。しかし、ただ速ければ良いというわけではない。スピードを上げ過ぎると、コースから外れて失格となってしまうからである。そのような点からミニ四駆ではスピードだけではなく、コースから外れない安定性も求められる。

また、特にコースアウトしやすい状況として、下り坂での加速によるものが多く見受けられる。したがって、今大会では下り坂での加速を制御し完走しやすくするための実装を行う。

2 実装

2.1 ハード

ArduinoNano を用いてミニ四駆の制御を行う。主な仕様については以下の通りである。

表 1: ArduinoNano の主な仕様

搭載マイコン	ATMEGA328P-AU
マイコン動作電圧	5V
ボード入力電圧	7-12V
フラッシュメモリ	32 キロバイト
SRAM	2 キロバイト
EEPROM	1 キロバイト
クロック周波数	16MHz

2.2 アルゴリズムと制御方法

ミニ四駆の速度の制御のアルゴリズムを以下に示す。ただし、コースによっては試合直前に変更する可能性がある。

[制御アルゴリズム]

1. 加速度センサにより車体の傾きを計算
2. 上り坂を検出した場合速度を抑える

ミニ四駆の速度の制御にはパルス副変調 (PMW)[1] という手法を用いる。パルス副変調とは、スイッチの ON と OFF を十分高速に繰り返し、その時間の比率を変更することで制御を行うという手法である。例えば、ON と OFF の時間の比率を 50 % にすることで出力を 50 % に抑えることが可能になる。

3 今後の課題

今回の制御方法は、坂道を検出し、制御を行うという単純な手法であった。そのため、坂道以外でコースアウトしてしまうようなコースだと対応できない。例えば、ストレートが続いた後にジャンプ台などが設置されていると、今回の手法では速度を落とさないままジャンプしてしまうためコースアウトする可能性が高くなる。

したがって、今後はコースアウトしそうな箇所を学習し、自動で制御するアルゴリズムを実装していきたいと考えている。

参考文献

- [1] Massimo Banzi 著、船田 巧 訳、”Arduino をはじめよう”、オライリー・ジャパン出版

ミニ四駆のためのモーター駆動制御

神津 隆大
電気通信大学

概要: ミニ四駆への Arduino と FET による制御回路の搭載による走行制御を図った。結果、モーター駆動のオンオフ制御を実現した。

1 ミニ四駆のカスタマイズ

使用したマシンと組み込んだパーツは、以下の通りである。

- マシン：ABILISTA(MA シャーシ)
- モーター：TORQUE-TUNED MOTER PRO
- モーター取付：モータークーリングシールド
- ターミナル：マシン付属のパーツ
- ギア・シャフト・ベアリング：マシン付属のパーツ
- ホイール：ワンウェイホイール
- タイヤ：ローハイトタイヤ
- ホイールシャフト：中空ステンレスシャフト
- ホイールベアリング：丸穴ボールベアリング
- シャーシ取付：MA シャーシ ファーストドライブパーツセット
- ブレーキ：マシン付属のパーツ
- ガイドローラー：HG 19mm オールアルミベアリングローラー

2 ボディの改造

CPU として Arduino Uno[1] を用いて制御を行うよう設計をし、機体に搭載、固定させるためボディ、シャーシの一部を切除した。走行中の安定性を考慮し、Arduino 基盤をリアウィングにネジ止めを施した。図 1 は、基盤を載せた機体の様子である。

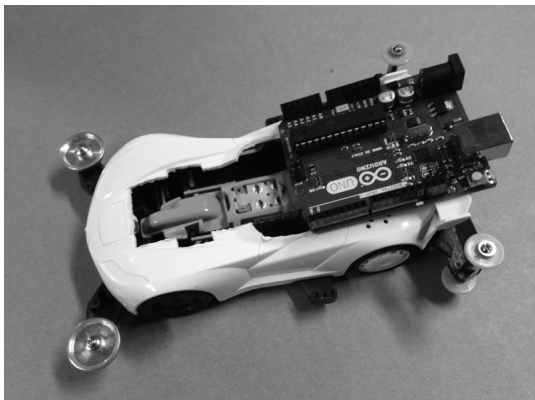


図 1: 基盤固定後の機体

3 制御方法

ミニ四駆は電池二本とモーターを接続して駆動しているが、ここに制御回路を挿入することにより、走行の制御を図った。今回はシーケンス制御の方針を取り、モーターのスイッチングとして p 型電界効果トランジスタ (MOSFET) を採用した。これにより、Arduino からの出力を MOSFET のゲートに接続することで、電流の制御が可能となる。

また、MOSFET ではオンオフ制御を行うため、通常の制御ではモーターの駆動を切か最大出力しか設定できない。そこで、ゲートへの入力にパルス波を扱うことにより、中間の出力を擬似的に生成することができる。これにより、中程度の速度といった制御が可能となる。

4 走行制御

Arduino に「5 秒間何も出力せず、その後 5 秒間 5V を出力せよ。これを繰り返せ。」といった記述をし、走行テストを行った。その結果、5 秒間加速し最高速で走り、その後 5 秒間加速をやめ、次第に停止していく様子が見れた。よって、Arduino によりモーター駆動のオンオフ制御が可能であることが確認できた。

次に、「2ms 間何も出力せず、その後 8ms 間 5V を出力せよ。これを繰り返せ。」といった記述をし、走行テストを行った。その結果、高速ではないゆっくりとした走行をする様子が見れた。よって、パルス波により中間の速度制御が可能であることが確認できた。

5 まとめ

ミニ四駆の走行にあたり、Arduino と MOSFET の導入による制御を図った。結果、モーター駆動の自動制御を実現した。

今後は機体に加速度センサーなどの機器を取り付け、坂道検出やコース形状の記憶のシステムを導入することにより、可変なコースに対しより柔軟な走行ができるよう努めたいと考える。

参考文献

- [1] Massimo Banzi(2012)「Arduino をはじめよう」船田巧訳, オライリー・ジャパン

手動編集によるミニ四駆 AI

原田 貴史

電気通信大学 情報理工学部 情報・通信工学科

概要

ミニ四駆はパーツを組み合わせることで、高速化と安定化をはかる。ミニ四駆 AI は AI を調整することで、高速化と安定化をはかるのだが、複雑なコースに対応できるような一般化された AI はなく、特殊な場合に対応する AI では全体の高速化が難しくなっている。そこで、本手法では人が操作することで AI を改善した。

1 目的

本手法では、人が操作してミニ四駆の制御することを目的とする。ミニ四駆 AI 大会ではセンサなどを使用してコースの形状を把握し、速度の調整をする AI が多く存在するが、小さなジャンプ台や直線後のジャンプ台などセンサなどで感知できない場合、感知しても速度が早すぎて制御が間に合わない場合があった。このような場合、人がプログラムを編集し特殊な状況に対応できるようにしていた。

そこで、本手法では人が速度制御をすることで高速で安定した走行を実現する。本番において操作ミスなどを起こさないよう、あらかじめテスト走行を行いそのデータを使用することでミスのない走行を実現する。

2 使用パーツ

ミニ四駆の制御ボード

-RT-AICHIP-V2[1]

モーター

-ハイパーダッシュモーター PRO

フロント

-AR シャーシ FRP フロントワイドステー

-二段アルミローラー

-フロントアンダーガード

リヤ

-AR シャーシ FRP リヤワイドステー

-13mm 低摩擦ブラローラー

3 AI

ミニ四駆 AI

(i) テスト走行時に人が走行速度を入力し、完走できるように調整をする。

(ii) 人の入力した速度と入力したタイミングをファイルに保存する。

(iii) 人の入力を元に AI に走行させる。

4 考察・まとめ

本手法では人が手動でミニ四駆を制御することでミニ四駆 AI の改善をはかった。

本手法によって特殊な状況にも応じることができ、同時に高速化をはかることもできる。入力データを用いることで、本番での入力ミスを防ぐことができる。

しかし、入力データを用いた場合、モータや電池の摩耗などハード面の変化に対応することができない。

電圧や走行速度の変化に伴い補正をかける AI などの補助を今後作成する必要がある。

参考文献

[1] " RT-AICHIP-V2 " ,<https://github.com/rt-net/AIchip>

加速度センサーによる速度制御を施したミニ四駆 AI

折原 レオナルド賢

電気通信大学大学院 情報システム学研究科

今回作成したミニ四駆では、加速度センサーから得られる加速度情報を元に、一定間隔で加算することで近似的な速度とし、速度制限を取り入れた。走行中、一定速度になると微小時間のモータ逆回転を行う。このモータ逆回転によって速度の減速を行い、コースアウトを未然に防ぐ。

1. 搭載した半導体

マイコンとして、Arduino Pro Mini 328 5V 16MHz。3軸加速度センサーモジュールとして LIS3DH を用いた。また、走行中のデータを送信するために XBee シリーズ 1 を用いた。

また、速度減速を行うためモータの逆回転を行っている。この制御のためにリレーを用いている。

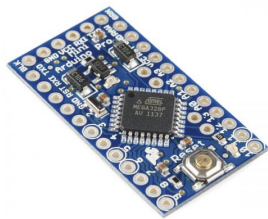


図 1 Arduino Pro Mini

2. 速度への換算

用いた加速度センサーでは 16bit で出力される。これを実際の加速度のデータに変換するために式 (1) を用いた。

$$a = x / (2^{14} * 9.8) \quad \dots (1)$$

式 (1) で得られる加速度より、100ms 間加算し続けることで近似的な速度としている。

3. モータの逆回転

速度がしきい値を超えた場合、減速が必要になる。このとき、モータを停止するだけでは減速が間に合わないことがあるため、微小時間モータを逆回転させることで減速を行う。

モータの逆回転はリレー回路を用いることによって、モータに入電する電流を反転させることで行う。

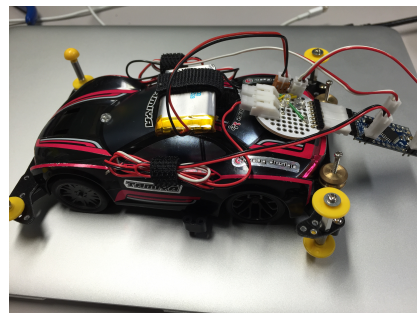


図 2 作成したミニ四駆

逆転回路ミニ四駆

GHITA ATHALINA

電気通信大学大学院 情報システム学研究科

今回作成したミニ四駆 AI は、加速度センサーとジャイロセンサーを用いて、制御が行われる。各センサーは基盤に載せ、Arduino と通信します。そして、読み取ったセンサー値で、スピードや角度の閾値を決め、プログラミングで制御する。

1. 使用するセンサー

1. 1 加速度センサー

利用する 3 軸加速度センサー

KXR94-2050 は XYZ 平面の 3 軸を検出することができる。今回は、X 軸の値を用いて、ミニ四駆の速度を制御する。

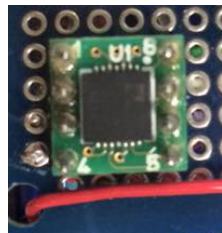


Fig1. 加速度センサー

1. 2 ジャイロセンサー

ジャイロセンサーは、回転を検知する方式により分類することができる。今回は、坂を登るときの閾値を求めするため GyroPitch という軸の値を読み取って、プログラムで制御する。



Fig2. ジャイロセンサー

2. 制御の仕組み

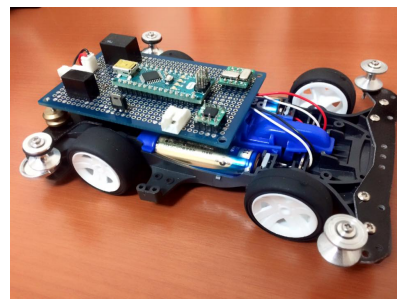


Fig3. 作成したミニ四駆

加速度センサーとジャイロセンサーの値を読み、この値をここでマイコンとして、Arduino のアナログピンに送信する。そして、送信されたデータを用いて、プログラムで制御し、また Arduino に書き込む。Arduino を動かすため、3.3V のリチウム電池が利用される。デジタルピンから出た信号で、モーターのスイッチが切り替わり、スピード原則や、逆回転することができる。

3. 参考文献

[1]<http://ednjapan.com/edn/articles/1406/09/news014.html>

[2]<http://d.hatena.ne.jp/licheng/20150407/p1>

PSoC 5 LP Prototyping Kitを用いたAIミニ四駆の制御

里見 優太

所属 高知工科大学 ロボット倶楽部

概要

cypress社が開発したPSoC5LP Prototyping Kitを用いて、ミニ四駆に搭載した6軸センサーから取得した値をもとに、コースの状態に応じてミニ四駆を制御する。

本文

1.ミニ四駆AI大会とは

これから人口の少なくなる私たちの国では、生活の場にある様々なものをより賢く使いこなすことがたいせつになる。そのための第一歩として安価で入手が容易であるミニ四駆をマイコン等を用いて制御することを目的とした大会である[1]。

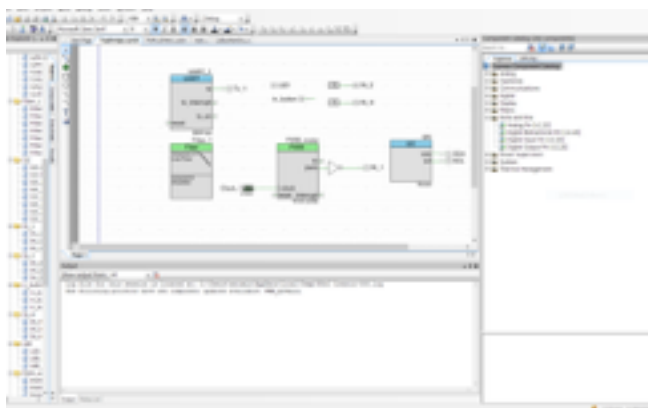
2.本体概要

私たちのチームは本体に「MAシャーシ」使用し、モータドライバやUARTを制御するためのマイコンとして「PSoC5 LP Prototyping Kit」を搭載している。

3.PSoCとは

PSoCとは、通常のマイコンとは違いデジタル回路とアナログ回路をIC内で設計できるマイコンである。そのため必要な機能を選択して使用することができる。また、内部でフィルタ処理などができることも強みの一つである。

機能の設定やプログラミングを書き込むのには統合設計開発環境の「PSoC Creator」を使用することによって、わかりやすく直感的に機能を追加することができる。[写真1]



[写真1 PSoC Creator]

4.制御仕様

制御にはI2C通信を用いて、6軸センサー(MPU6050)から加速度とジャイロの値を取得する

取得した値をノイズフィルタにかけ、ミニ四駆が走行しているコースの形状や、ミニ四駆の傾きを求めてPWMで走行速度の制御を行う。

参考文献

- [1] “AI & mini4WD” 「はじめに」 ,
< http://ai4wd.org/?page_id=2 > 2016年2月25
日アクセス

ターン制戦略ゲーム探索のこのへんファジィ評価を用いた効率化

武藤 孝輔 西野 順二
電気通信大学

概要: 1つのユニット行動ごとに分解したゲーム木を構成してノードごとの分岐を減らしUCT探索を行う。人間の直感に基づいた判断によるノードの探索優先度の順位付けを行うためにファジィ評価を用いる。

1 M-UCT

ユニット順を考慮しつつ探索する M-UCT を提案する。M-UCT とは、ユニット単位で展開する探索木を生成し、その探索に UCT 探索 [1] を用いたものである。UCT 探索は、ゲーム木のノード評価にモンテカルロシミュレーションを用い、有望なノードを逐次的に展開することで、より精度の高い確率的探索を行う手法である。とくに囲碁の人工プレイヤーで高い効果を上げており [2]、ターン内の合法手が多いゲームである Amazons にも高い効果を示している [3, 4]。

1.1 M-UCT のアルゴリズム

M-UCT

- (i) 局面の状態 s を根ノードの盤面の状態とする
- (ii) s における未行動のユニットの行動の集合 $A(s)$ に含まれる a による次局面 $s'(s, a)$ が、 s の子ノードとなる
- (iii) $A(s)$ から UCB 値が最大の行動 a^* の枝をたどり、子ノードに向かう

$$UCB = X_i + c\sqrt{\frac{\log N}{n_i}} \quad (1)$$

$$a^* = \operatorname{argmax}_{a \in A(s)} UCB(a) \quad (2)$$

- (iv) 選択された子ノードが末端でなければ、 $s = s'(s, a^*)$ として (2) に戻る
- (v) たどり着いた末端ノードのプレイアウト回数が閾値を超えていた場合、1 段展開し (2) に戻る
- (vi) プレイアウト回数が閾値を超えていない末端ノードにたどり着いたらプレイアウトを行う
- (vii) プレイアウト結果をたどってきたノードに反映する

2 ゲーム局面の大局的ファジィ評価を用いた F-UCT 探索

分岐因子が大きく探索が困難な戦略ゲームに対しては、モンテカルロ探索の有効性も限定的である。その一

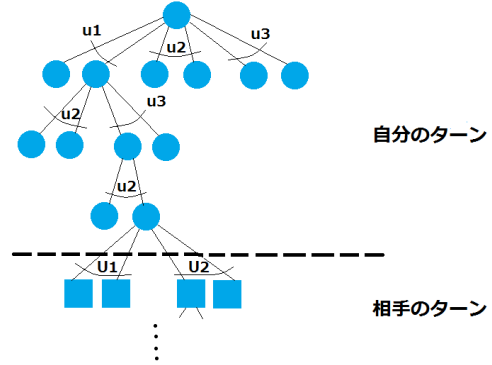


図 1: M-UCT の探索木

方で人間プレイヤーはそれなりに強く、コンピュータプレイヤーに対しても勝ち越す現状である。このとき、人間プレイヤーがコンピュータより深く広い探索を行っていることはほぼあり得ず、直感的な局面判断に優れ、大幅な探索の枝刈りができているためと考えられる。

この、人が直感的に行うゲーム局面の大局的な評価をファジィ集合を介して行うファジィ評価を提案する。

ファジィ集合は人のもつ概念をある要素がどれだけその集合に含まれるかという帰属度（メンバーシップ値）によって特徴付けて、あいまいな集合として定式化したものである。ゲーム局面の評価においては、人の知識にもとづいて良い盤面というあいまいな集合を定義し、ゲーム中、新たに得られた局面の良い盤面集合への帰属度を参照することで人の直感にもとづいたその局面の良さを評価することができる。

2.1 ファジィ集合

ファジィ集合とはファジィ理論の根幹をなす概念で、大きな数のような境界のあいまいな集合である。Zadeh はこのようなファジィ集合 A を全体集合 X の上で、式 (3) に示すメンバーシップ関数 μ_A を定義することで特徴付けた。

$$\mu_A : X \rightarrow [0, 1] \quad (3)$$

たとえばある要素 $x \in X$ があいまいな集合 A に含ま

れる度合いは $\mu_A(x) = 0.8$ などとして表現される。

2.2 ファジィ集合の同定と生成

ゲーム局面の状態空間は多変数で離散的なため、人手によったり、一次元のファジィ数をもとにすることは困難である。ある局面がどのように分類されるかについては、人の直感的な判断を得る事ができるので、データに基づいたファジィ集合の生成を行う。

たとえば特徴量ベクトル v_i の空間 V 上で、「良い局面」というファジィ集合 G を生成することを考える。

以下の手順で生成することができる。

ファジィ集合の生成

- (i) 局面 s_i を一つ生成する
- (ii) 人間プレイヤーが G に含まれると言えるかどうか判断し $h(s_i) \in \{0, 1\}$ として含まれるかどうかを記録する
- (iii) 十分なサンプル集合 $S = \{s_1, s_2, \dots, s_n\}$ が集まるまで 1 へ戻る
- (iv) 得られたサンプル s_i を特徴量ベクトル $v_i = v(s_i)$ に変換する
- (v) データサンプルの特徴量空間での相対密度を各サンプル点で計測する。サンプル点特徴量 v_i について標準距離 D_ε 以内の点集合 $N_i = \{v_j | d(v_i, v_j) \leq D_\varepsilon\}$ の要素数を数え、その最大値 $Max_N = \max(|N_i|)$ に対する割合 $\mu_G(v_i) = |N_i|/Max_N$ を v_i の帰属度とする。ここで $d()$ はユークリッド距離関数である。
- (vi) すべての $v_i \in V$ についての帰属度を表 T_G として保持する

実装上は、ここで生成された帰属度の表 T_G がファジィ集合 G のメンバーシップ関数の実体である。表に含まれない一般の特徴量ベクトル $v \in V$ については値の定まった要素から式 (4) に示す k 近傍集合 K からの重み付き平均補完によって求める。

$$\mu_G(v) = \frac{\sum_{v_i \in K} w_i \mu(v_i)}{\sum_{v_i \in K} w_i} \quad (4)$$

ここで v の k 近傍は、 T_G 中で v への距離が近い順に k 個の v_i からなる集合であり、 w_i は距離の逆数 $w_i = 1/d(v_i, v)$ である。

3 F-UCT

M-UCT にファジィ判断の評価による探索優先度の順位付けを加え、これを F-UCT とする。ユニットの種類毎に生成されたファジィ集合には、表??により分類される各行動ベクトルにメンバーシップ値が与えられてい

る。このメンバーシップ値を探索優先度の重みに用い、ユニット行動木において新しく展開された子ノードの中からメンバーシップ値の高い行動を優先して探索する。

F-UCT

- (i) 局面の状態 s を根ノードの盤面の状態とする
- (ii) s における未行動のユニットの行動の集合 $A(s)$ に含まれる a による次局面 $s'(s, a)$ が、 s の子ノードとなる
- (iii) ファジィ集合から $s'(s, a)$ のメンバーシップ値を求める
- (iv) $s'_i = s'(s, a_i)$ に対し、 s'_i の訪問回数 n_i をメンバーシップ値 $times10$ 、報酬 X_i を 1 とする
- (v) M-UCT アルゴリズムに従う

「このへんファジィ」は西野順二の登録商標です。本研究の一部は科学研究費補助金基盤研究 C(26330273) により支援されたものである。

参考文献

- [1] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton. A survey of monte carlo tree search methods. *Computational Intelligence and AI in Games, IEEE Transactions on*, Vol. 4, No. 1, pp. 1–43, march 2012.
- [2] M. Enzenberger, M. Muller, and R. Segal B. Arneson. Fuego-an open-source framework for board games and go engine based on monte carlo tree search. *Computational Intelligence and AI in Games, IEEE Transaction on*, Vol. 2, pp. 259–270, 2010.
- [3] Julien Kloetzer, Hiroyuki Iida, and Bruno Bouzy. The monte-carlo approach in amazons. *Computer Game Workshop 2007*, 2007.
- [4] Lorentz Richard J. Amazons discover monte-carlo. *Computers and Games 2008*, 2008.

問い合わせ先：

〒182-8585 調布市調布ヶ丘 1-5-1
電気通信大学 情報・通信工学専攻
武藤孝輔
TEL/FAX: 042-443-5578

深層学習を用いたデジタルカーリングプログラム Deep Imitation System の提案

益子 直[†] 成澤 和志^{††} 篠原 歩^{††}

[†] 東北大学 工学部 〒980-8579 宮城県仙台市青葉区荒巻字青葉 6-6-05

^{††} 東北大学 大学院情報科学研究科 〒980-8579 宮城県仙台市青葉区荒巻字青葉 6-6-05

E-mail: [†]sunao_mashiko@shino.ecei.tohoku.ac.jp, ^{††}{narisawa, ayumi}@ecei.tohoku.ac.jp

要約 本稿では、深層学習を用いたデジタルカーリングプログラム **Deep Imitation System (DIS)** を提案し、実験により強さの確認を行う。実験の結果、DIS は Take という別のデジタルカーリングプログラムと同程度の実力を持つことが分かった。

1. はじめに

深層学習は多層のニューラルネットワークを用いた機械学習の総称である [3]。深層学習は画像認識において高い性能を発揮し [1]、注目を浴びている。また、深層学習はゲームが強い人工知能を実現する学習方法としても期待されている [4]。

本稿では、深層学習を用いて強いプレイヤーのショットを模倣するデジタルカーリングプログラム、**Deep Imitation System (DIS)** を提案する。また、DIS の強さを確認するために他のデジタルカーリングプログラムと対戦させる実験を行う。

2. 深層学習

多層のニューラルネットワークを用いた機械学習のことを総称して**深層学習**と呼ぶ。深層学習では多次元の入力から多次元の出力を計算する**モデル**を作成する。モデルには**ユニット**が層状に配置されており、入力に一番近い層を**入力層**、出力に一番近い層を**出力層**、入力層と出力層の間にあるいくつかの層を**中間層**または**隠れ層**と呼ぶ。各ユニットは**重み**、**バイアス**と呼ばれる2つのパラメータを持ち、さらに**活性化関数**と呼ばれる関数を持つ。

モデルに訓練データとして入力と目標出力の組を与えると、モデルの出力と目標出力の**損失**が計算される。計算された損失をもとに、モデルは各ユニットの重みとバイアスの値を損失が小さくなるように更新する。モデルに対し訓練データを大量に用意することにより、モデルは損失が最小限となるような重みとバイアスを

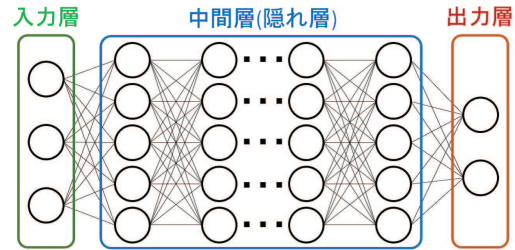


図1 多層パーセプトロン

学習をしていく。

モデルにはいくつかの種類があるが、本稿では多層パーセプトロンを採用する。多層パーセプトロンを図1に示す。

3. Deep Imitation System

深層学習モデルを用い、強いプレイヤーのショットを模倣するようなデジタルカーリングプログラム **Deep Imitation System (DIS)** を提案する。DIS は、ゲームの盤面を入力とし、盤面に適したショットが出力されるような深層学習のモデルを作成する。DIS の学習では、ショットの各パラメータの大きさの違いを考慮し、各パラメータに**パラメータ優先度**を設ける。訓練データの目標出力であるショットの各パラメータ値にパラメータ優先度を掛け合わせてから学習を行い、実際にモデルを使用するときには出力されたショットの各パラメータ値をパラメータ優先度の値で割る。パラメータ優先度の導入により、各パラメータの学習度合を任意に変更することが可能になると考えられる。

表 1 chainer の設定

設定項目	本実験での設定	備考
optimizer	RMSpropGraves	各パラメータ：デフォルト値
入力層ユニット数	35	活性化関数：なし
中間層ユニット数	500 × 3 層	活性化関数：tanh
出力層ユニット数	3	活性化関数：なし
誤差関数	平方二乗誤差	
訓練データの学習	ミニバッチ学習	バッチサイズ：128

表 2 深層学習に用いた入出力

層	種類	ユニット数
入力層	ストーン x 座標	16
	ストーン y 座標	16
	現在エンド数	1
	現在ストーン数	1
	現在スコア差	1
	合計入力ユニット数	35
出力層	x 軸初速度	1
	y 軸初速度	1
	回転方向	1
	合計出力ユニット数	3

4. 実験

DIS の実力を調べるため、他のプログラムと実際に対戦させる実験を行う。対戦相手は第 2 回 UEC 杯に出場した 6 つのプログラムとし、各プログラムに対し先攻後攻を入れ替えながら 10 試合ずつ、計 60 試合の対戦を行う。カーリングシミュレータのパラメータは GAT 杯のものに準拠するが、第 2 回 UEC 杯とは持ち時間が異なり、必ず時間切れになってしまうプログラムが存在したため、各持ち時間は 438 秒として行うものとする。また、引き分け時の延長戦は行わないものとする。

本稿における実験では、DIS における x 軸初速度、 y 軸初速度、回転方向のパラメータ優先度はそれぞれ 30, 1, 3 として行う。DIS の訓練データは、強さとデータ収集のしやすさという観点から、デジタルカーリングプログラム **GCCS** のショットを用いる。訓練データは、試合時間を 438 秒とし、GCCS と第 2 回 UEC 杯に出場した 6 つのプログラムを先攻後攻を入れ替えながら 50 試合ずつ、計 250 試合戦わせることにより得る。得られた GCCS の全ショット 16000 投のうち、14080 投を訓練データ、1920 投を評価データとして使用し、10000 回学習を行う。

また、本実験では、OS が Windows 7 Professional, CPU が Intel Xeon CPU W3565 @ 3.20GHz, コア数は 8, メモリサイズ 12GB の計算機を使用する。深層学習プログラムの実装には、**chainer** [2] と呼ばれる深層学習ライブラリを使用する。chainer における設定を表 1 に示す。本実験に用いたモデルの入出力を表 2 にまとめる。対戦の結果を表 3 に示す。

表 3 他プログラムに対する DIS の対戦結果

対戦相手	勝利数	敗北数	引分数	勝率 [%]	平均得失点差
GCCS	0	10	0	0	-11.8
歩	0	10	0	0	-8.7
じりつくん	0	10	0	0	-26.2
Ramen	0	10	0	0	-21.8
KC002	0	10	0	0	-18.0
Take	4	4	2	50	-0.3

5. おわりに

本稿では、深層学習を用いたデジタルカーリングプログラム Deep Imitation System (DIS) を提案した。また、DIS に対して実験を行い、デジタルカーリングプログラムの実力としては Take というプログラムと同程度の実力を持つことが分かった。

今後の課題としては、次の 2 つが考えられる。1 つは、訓練データ数、評価データ数を増やすことである。深層学習において、学習データが多いほど精度が増す傾向があるため、誤差は小さくなると考えられる。2 つ目は、ショット数で学習の優先度を設けることである。1 エンドにおいて、序盤のショットは石が少ないために学習が簡単だと考えられる。逆に、終盤のショットは石が多くなるために盤面が複雑になりやすく、学習は困難になると考えられる。また、カーリングにおいては、終盤のショットは点数に直結することが多く、1 回のショットの価値が高い傾向にある。よって、序盤の学習の優先度を下げ、終盤の学習の優先度を上げることでより良いショットが行えるのではないかと考えられる。

また、本稿における提案手法では強いプログラムの模倣で留まってしまうため、深層学習でのモデルの学習が終了した後、更にモデルの学習を進める手法の検討が考えられる。さらに、別の手法として、**Deep Q Network** をデジタルカーリングに対し適用することが考えられる。

文 献

- [1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pp. 1097–1105, 2012.
- [2] Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. Chainer: a next-generation open source framework for deep learning. In *NIPS*, 2015.
- [3] 岡谷貴之. 深層学習. 講談社, 2015.
- [4] 小林雅一. AI の衝撃 人工知能は人類の敵か. 講談社, 2015.

デジタルカーリングにおける戦略的知識を用いた候補手の絞り込み

城戸 皓己 伊藤 毅志

電気通信大学 大学院情報理工学研究科 情報・通信工学専攻

概要: 多くのデジタルカーリング AI はモンテカルロ法を適用している。モンテカルロ法を効率化するためにはプレイアウトの質を高めることが有効である。そこで、本報告では戦略的知識に基づいて、状況に応じて候補手を絞り込む手法を提案し、検討した。

The narrowing-down of candidate moves with strategic knowledge in digital curling

Koki KIDO Takeshi ITO

Department of Communication Engineering and Informatics,
The University of Electro-Communications

Abstract A lot of digital curling AI applies Monte Carlo method. It is proved to be effective to streamline Monte Carlo method by enhancing playout's quality. Therefore, in this report, depending on strategic knowledge, we proposed and examined an approach which can narrow down candidate moves by considering the situation.

1. はじめに

北清らによって提唱された「デジタルカーリング」により、カーリングの不確定性をコンピュータ上で表現することによって、カーリングの戦略を議論する場が提供され、大会も開かれるようになっていく。これにより、近年カーリングは、ゲーム AI の研究テーマの一つとして定着しつつある[1]。

カーリングは、適当な候補手を選び続けても終了することが保証されていること、評価関数の設計が難しいことなどからモンテカルロ法を組み込んだ AI が多い。モンテカルロ法を効率化するためには、一つ一つのプレイアウトの質を高めることが有効であることが指摘されている。プレイアウトの質が強さを向上させることを示す研究として、Gelly らは 3×3 のパターンの囲碁の知識を利用することでプレイアウトの質を高める手法を提案し、ランダムプレイアウトより強いコンピュータ囲碁プログラムを実現した[2]。また、カードゲーム大貧民においても、プレイアウトの質を高めることの重要性を示す研究がある[3]。

プレイログの入手が比較的容易なゲームでは、それをもとに機械学習でプレイアウトの質を高める手法が効果を見せているが、カーリングは電子的なプレイログが十分に存在しないばかりか、その記録形式すら整備されていない。そこで本研究では、カーリングの戦略書に記載されている戦略的知識に基づいて、状況に応じて候補手を絞り込む手法を提案する。評価実験では、どの程度候補手を絞り込めるのか、どの程度強くなる可能性があるのかについて議論する。

2. カーリングの基本戦略

2.1 カーリングの戦略的知識

カーリングは、戦略的側面だけでなく、技術的側面も勝敗に影響するため、戦略の議論が難しく、戦略はコーチなどによる口伝で伝わる傾向が強い。そのため、戦略を論じる専門書は少ない。その中で、「Introduction to Curling Strategy」は戦略を体系的

にまとめているほぼ唯一の文献である。ここでは、この文献（以降、「戦略書」と呼ぶ）に基づいて説明していく[4]。

2.2 カーリングの基本的な考え方

カーリングは後攻が得点をする可能性が大きいことから、後攻がアドバンテージを持つゲームである。基本的な戦略として先攻は次のエンドで後攻を得るために後攻に 1 点だけを取らせることを目指す。逆に後攻は 2 点以上をとるか、後攻を維持するためにブランクエンド(どちらのチームも得点しないエンド)にすることが基本戦略となる。

2.3 エンドプラン

エンドプランとは、エンドにおける得点の目標である。エンドプランは点差・先攻後攻・エンド数によって決定される。

エンドプランは攻撃的・保守的に分類できる。攻撃的・保守的の違いはエンド目標を達成するまでにどれだけリスクを許容するかに応じて決められる。攻撃的プランはリスクを負ってエンド目標を達成するプラン、保守的プランはリスクを少なくしながらエンド目標を達成するプランのことである。

3. エンドプランを導入したシステム

3.1 候補手の生成

戦略書によると、点差、先攻後攻、エンド数に応じて、大別して以下の 6 つのエンドプランに分類される。

ー先攻 (Steal, Force1, Prevent Multi)

ー後攻 (Score 3 or more, Score2, Prevent Steal)

先攻のエンドプランは先攻が得点を取る Steal、後攻に 1 点だけ得点させる Force 1、後手に複数点を取られるのを回避する Prevent Multi がある。

後攻のエンドプランは 3 点以上を得点する Score 3 or more、2 点を得点する Score 2、先手に Steal させないことだけを注目する Prevent Steal がある。

まず、想定されるエンド数と得点差、及び先攻後攻に応じて、それぞれのエンドプランが用いられるの

かを表にした。さらに、それぞれのエンドプランに応じて、候補手（センターガード3種、カムアラウンド3種、フリーズ3種、ヒットアンドステイ、ランバック、スプリット、ドローショット4種、ガードショット6種類、ティーショット、タップショット、パスショット）を局面状況に応じて分類した。

これを図示したものが、図1である。エンド数・得点差・先攻後攻の情報を、戦略書に基づいて作成した表から6種類のエンドプランの1つが選ばれる。そのエンドプランに応じて、局面のストーンの配置からショットの種類が絞られ、それに応じた候補手が生成される。

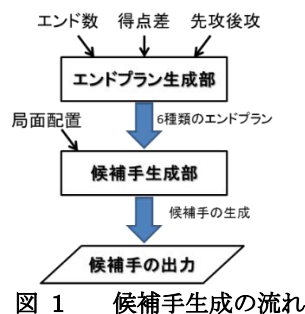


図1 候補手生成の流れ

3.2 提案システム

提案システムは、第1回 UEC 杯デジタルカーリング大会で優勝した「歩」をベースに作成した [5]. 「歩」も作成者によるヒューリスティクスを組み込んで候補手を絞り込んでいるが、体系的な知識の導入は行っていないため絞り込みは十分ではない。この候補手生成部分に上記で提案した候補手生成を導入することで強化をはかる。

「歩」はモンテカルロ木探索を使用しており、大会でも実績のあるプログラムであるので、本研究のベースとするプログラムとして最適であると考えた。

「提案システム」の先攻のエンドプランの決定方法を表1に示した。後攻のエンドプランの決定方法は付録Aに掲載する。例えば、エンド数が8、得点差が2の場合はエンドプラン Force 1 が選択される。

表1 提案システムの先攻のエンドプランの決定方法

先攻から見た得点差								
先攻	-3以下	-2	-1	0	1	2	3	4以上
1～7エンド	Steal			Force 1				Prevent Multi
8エンド	Steal or Force 1					Force 1		
9エンド	Steal				Force 1 or Steal		Force 1 or Prevent Multi	
10エンド	Steal					Force 1	Prevent Multi	

表2は「提案システム」のエンドプラン Force 1 における候補手生成を表したものである。例えば、センターガードのみの場合はカムアラウンドショットが選択される。表2の※1は No.1 ストーンが自分のチームではない場合に候補手に加え、※2は No.1 スト

ーンが自分のチームである場合に候補手に加えることを指す。候補手生成の表をエンドプランごとに設計した。その他のエンドプランの候補手生成の表は付録Aに掲載する。

表2 提案システムのエンドプラン Force 1 の候補手生成

条件			ショットの種類						
センターガードがあるかどうか?	リンク内に相手のストーンがあるかどうか?	ハウス内に自分のストーンがあるかどうか?	センターガード(2種類)	カムアラウンド(3種類)	フリーズ(3種類)※1	ヒットアンドステイ	ランバック	スプリット	ドローショット(4種類)※2
x	x	x	○	○					
x	x	○	○		○			○	○
x	○	x	○	○		○	○		
x	○	○	○		○	○	○	○	○
○	x	x		○					
○	○	x		○		○			
○	x	○			○			○	○
○	○	○			○	○	○	○	○

3.3 評価実験

提案手法を実装した AI が従来の「歩」より強くなっているかを検証するために、提案手法を何投目まで適用するかで、16投目まで、12投目まで、8投目まで、4投目までの4通りに分けたプログラムを用意した。それぞれ、プログラムA、プログラムB、プログラムC、プログラムDとした。これらのプログラムと従来の「歩」との対戦実験を行った。持ち時間は1手1秒とし、それぞれ先後50試合ずつ、計100試合ずつ行った。

3.4 結果

4つのプログラムの従来の「歩」に対する勝率を表3に示す。勝率は引き分けを除いて計算した。また、図2にプログラムAの平均候補手数を示す。

表3 提案システムの「歩」に対する勝率(%)

	勝率
プログラムA	24.42
プログラムB	47.50
プログラムC	45.68
プログラムD	51.22

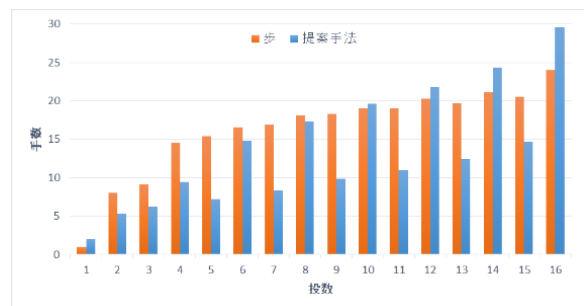


図2 プログラムAの平均候補手数

3.5 考察

プログラムAが大きく負け越す結果となった。原因としては、戦略書には序中盤の記述があるが、終盤の知識の記述が少ないことや後攻の候補手の絞り込みができていないため大きく負け越したと考えられる。

また、中盤を見ると、十分な手の絞り込みがなされ

ていない様子も観察された。更なる候補手の絞り込みが必要であることが示唆された。

4. 改良システム

4.1 攻撃的、保守的プランの導入

前章で導入したエンドプランのうち、先攻は、Steal, Force1, 後攻 Score2 について、攻撃的、保守的プランを導入して更に細分化し、候補手をさらに絞り込むことにした。この候補手を「歩」に導入した「改良システム」を提案した。

「改良システム」の先攻のエンドプランの決定方法を表 4 に示した。赤色が攻撃的、青色が保守的、白色がどちらでもないことを表している。後攻のエンドプランの決定方法については付録 B に掲載する。例えば、エンド数が 8、得点差が 2 の場合はエンドプラン攻撃的な Force 1 が選択される。

表 4 改良システムの先攻のエンドプランの決定方法

	先攻から見た得点差							
先攻	-3以下	-2	-1	0	1	2	3	4以上
1～7エンド	Steal	Steal	Force 1			Force 1		Prevent Multi
8エンド	Force 1					Force 1		Force 1
9エンド	Steal				Steal	Force 1	Force 1	
10エンド	Steal				Steal	Force 1	Prevent Multi	

表 5 は「改良システム」のエンドプラン攻撃的な Force 1 における候補手生成を表したものである。表 5 の※は相手のストーンがティーラインより後ろの場合に候補手に加えることを指す。候補手生成の表をエンドプランごとに設計した。その他のエンドプランの候補手生成の表は付録 B に掲載する。

表 5 改良システムのエンドプラン攻撃的な Force 1 の候補手生成

条件			ショットの種類						
センターガードがあるかどうか？	リンク内に相手のストーンがあるかどうか？	ハウス内に自分のストーンがあるかどうか？	センターガード(2種類)	カムアラウンド	フリース※	ヒットアンドステイ	ランバック	ガードショット(2種類)	スプリット
x	x	x	○						
x	x	○	○					○	
x	○	x	○		○	○	○		
x	○	○	○		○	○	○	○	
○	x	x		○					
○	○	x		○	○	○	○		
○	x	○		○	○	○	○	○	
○	○	○		○	○	○	○	○	

4.2 評価実験

改良システムが従来の「歩」より強くなっているかを検証するために、改良システムを用いた 3 つのプログラム（12 投目まで：プログラム B'，8 投目まで：プログラム C'，4 投目まで：プログラム D'）を用意し、従来の「歩」との対戦実験を行った。持ち時間は 1 手 1 秒とし、それぞれ先後 250 試合ずつ、計 500 試合ずつ行った。

4.3 結果

3 つのプログラムの従来の歩に対する勝率を表 6 に

示す。勝率は引き分けを除いて計算した。図 3 にプログラム B'での平均候補手数を示す。

表 6 改良システムの「歩」に対する勝率(%)

	勝率
プログラムB'	47.07
プログラムC'	53.92
プログラムD'	51.35

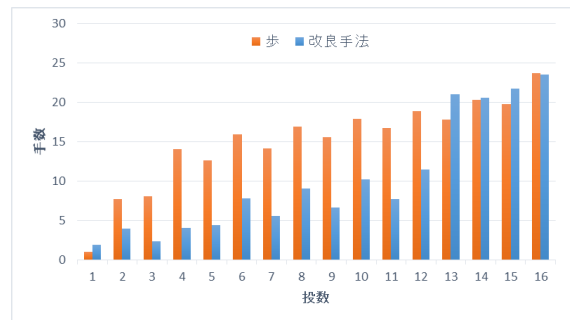


図 3 プログラム B' の平均候補手数

4.4 考察

プログラム C' では、二項検定で p 値が 0.051>0.05 より有意差は無かったものの勝率向上について有意傾向が見られた。

平均候補手数に関しては、全体的に従来の AI より絞り込めていることが確認された。

5. おわりに

本研究では、知識を用いて候補手を絞り込む手法を提案した。残念ながら有意に強くなるプログラムは作れなかったが、適切な知識を導入することで適切に候補手の絞り込みを行うことができ、強くなる可能性が示唆された。

有効な手の絞り込みができていのかどうかを候補手の採用率などから検証することで、より適切な知識への改良について考慮したい。

参考文献

- [1] Takeshi Ito and Yuuma Kitasei: Proposal and Implementation of “Digital Curling”, 2015 IEEE Conference on Computational Intelligence and Games (CIG2015), pp.469-473 (2015).
- [2] Sylvain Gelly, Yizao Wang, Remi Munos, Olivier Teytaud: Modification of UCT with Patterns in Monte-Carlo Go, [Research Report] RR-6062, INRIA, (2006).
- [3] 地曳隆将, 松崎公紀, “大貧民における棋譜データからの提出手役評価関数の学習”, 研究報告ゲーム情報学 GI-31, No. 15, pp. 1-7, (2014).
- [4] Gabrielle Coleman, Introduction to Curling Strategy, ISBN 9781941164020, (2015).
- [5] 「第 1 回 UEC 杯デジタルカーリング大会」, <<http://minerva.cs.uec.ac.jp/curling/wiki.cgi>>, (参照 2016-02-01).

付録

付録 A 提案システムに関する表

表 A.1 後攻のエンドプランの決定方法

後攻から見た得点差								
後攻	-3以下	-2	-1	0	1	2	3	4以上
1～7エンド	Score 3 or more	Score 2						
8エンド	Score 3 or more, Score 2	Score 2						
9エンド	Score 3 or more	Score 3 or more, Score 2	Score 2				Score 2 or Prevent Steal	
10エンド	Score 3 or more	Score 2				Prevent Steal		

表 A.2 エンドプラン Steal の候補手生成

条件			ショットの種類										
センターガードがあるかどうか？	リンク内に相手のストーンがあるかどうか？	ティー付近にストーンがあるかどうか？	センターガード(4種類)	ハイセーターガード	フリーズ(3種類)	タップショット	カムアラウンド(3種類)	ティーショット	ドローショット(2種類)	サイドショット(2種類)	ガードショット(6種類)	ヒットアンドステイ	ランバック
x	x	x	o			o	o	o					
x	o	x	o			o	o	o				o	o
x	x	o	o			o	o	o					
x	o	o	o			o	o	o				o	o
o	x	x		o					o		o		
o	x	o	o						o		o		
o	o	x	o							o		o	o
o	o	o	o								o		

表 A.3 エンドプラン Prevent Multi の候補手生成

条件		ショットの種類			
リンク内に相手のストーンがあるかどうか？	相手のストーンがティーラインより後ろかどうか？	パスショット	ヒットアンドステイ	ランバック	フリーズ
x	x	○			
x	○	○			
○	x		○	○	
○	○		○	○	○

表 A.4 の※1 はストーンがティーラインより後ろにある場合に候補手に加え、※2 はラストストーンの場合に候補手に加えることを指す。

表 A.4 エンドプラン Score 2 の候補手生成

条件		ショットの種類									
コーナーガードがあるかどうか？	リンク内に相手のストーンがあるかどうか？	ハウス内に自分のストーンがあるかどうか？	コーナーガード(4種類)	カムアラウンド(3種類)	フリーズ(3種類)※1	ヒットアンドステイ	ランバック	スプリット	ドローショット(4種類)※2	ガードショット(6種類)	レイズショット
x	x	x	o								
x	x	o	o							o	o
x	o	x	o	o	o	o	o				
x	o	o	o	o	o	o	o			o	o
o	x	x								o	o
o	x	o								o	o
o	o	x		o	o	o	o				
o	o	o		o	o	o	o	o		o	o

表 A.5 の※1 は打球数が 14 投目と 16 投目の場合に候補手に加えることを指す。

表 A.5 エンドプラン Score 3 or more の候補手生成

条件			ショットの種類							
コーナー ガードがある かどうか？	リンク内に相 手のストー ンがあるか どうか？	ハウス内に 自分のストー ンがあるか どうか？	セン ター ガード (4種類)	カムア ラウンド (3種類)	フリー ズ(3種類)	ヒット アンド ステイ	ラン バック	ドロー ショット (4種類) ※1	ガード ショット (6種類)	タップ ショット
x	x	○	○		○				○	○
x	○	○	○	○	○	○	○			
x	○	x	○	○	○	○	○		○	○
○	x	x								
○	x	○			○				○	○
○	○	x		○	○	○	○		○	○
○	○	○		○	○	○	○		○	○

表 A.6 エンドプラン Prevent Steal の候補手生成

条件		ショットの種類			
リンク内に相手のストーンがあるかどうか？	16投目であるかどうか？	パスショット	ヒットアンドステイ	ランバック	ティーショット
x	x	○			
x	○				○
○	x		○	○	
○	○		○	○	○

付録 B 改良システムに関する表

表 B.1 後攻のエンドプランの決定方法

	後攻から見た得点差								
後攻	-3以下	-2	-1	0	1	2	3	4以上	
1～7エンド	Score 3 or more	Score 2		Score 2					
8エンド	Score 3 or more, Score 2	Score 2		Score 2					
9エンド	Score 3 or more	Score 3 or more, Score 2	Score 2	Score 2			Score 2 or Prevent Steal		
10エンド	Score 3 or more	Score 2		Score 2		Prevent Steal			

表 B.2 の※は相手のストーンがティーラインより後ろの場合に候補主に加えることを指す。

表 B.2 エンドプラン保守的な Force 1 の候補手生成

条件		ショットの種類							
センターガードがあるかどうか？	リンク内に相手のストーンがあるかどうか？	ハウス内に自分のストーンがあるかどうか？	センターガード(2種類)	カムアラウンド	フリーズ※	ヒットアンドステイ	ランバック	ガードショット(2種類)	スプリット
x	x	x	o						o
x	x	o	o						
x	o	x	o		o	o	o		
x	o	o	o			o	o		o
o	x	x		o		o	o		
o	o	x			o	o	o		
o	x	o		o		o	o		o
o	o	o		o	o	o	o		o

表 B.3 エンドプラン攻撃的な Steal の候補手生成

条件			ショットの種類						
センターガードがあるかどうか？	ティーラインに相手のストーンがあるかどうか？	リンク内に相手のストーンがあるかどうか？	センターガード(2種類)	ハイセーターガード	ティーショット	スプリット	ガードショット	ヒットアンドステイ	ランバック
x	x	x	o						
x	x	o	o					o	o
x	o	x	o			o	o		
x	o	o	o				o	o	o
o	x	x		o	o				
o	x	o			o			o	o
o	o	x		o		o	o		
o	o	o		o	o	o	o	o	o

表 B.4 エンドプラン保守的な Steal の候補手生成

条件			ショットの種類						
センターガードがあるかどうか？	ティーラインに相手のストーンがあるかどうか？	リンク内に相手のストーンがあるかどうか？	センターガード(2種類)	ハイセーターガード	ティーショット	スプリット	ガードショット	ヒットアンドステイ	ランバック
x	x	x	o						
x	x	o	o					o	o
x	o	x	o			o			
x	o	o	o			o		o	o
o	x	x			o				
o	x	o			o			o	o
o	o	x				o			
o	o	o				o	o	o	o

表 B. 5 エンドプラン Prevent Multi の候補手生成

条件			ショットの種類			
リンク内に相手のストーンがあるかどうか？	相手のストーンがティーラインより後ろかどうか？		パスショット	ヒットアンドステイ	ランバック	フリーズ
×	×		○			
×	○		○			
○	×			○	○	
○	○			○	○	○

表 B. 6 エンドプラン攻撃的な Score 2 の候補手生成

条件			ショットの種類								
コーナーガードがあるかどうか？	リンク内に相手のストーンがあるかどうか？	リンク内に自分のストーンがあるかどうか？	コーナーガード (4種類)	カムアラウンド (5種類)	フリーズ※ (3種類)	ティーライン上にドローショット※	タップショット	ヒットアンドステイ	ランバック	ガードショット	スプリット
×	×	×	○								
×	×	○	○							○	○
×	○	×	○		○			○	○		
×	○	○	○		○			○	○	○	○
○	×	×		○							
○	○	×		○	○			○	○		
○	×	○		○						○	○
○	○	○		○	○			○	○	○	○

表 B.7 の※は相手ストーンがティーラインより後ろの場合に候補手に加えることを指す。

表 B. 7 エンドプラン保守的な Score 2 の候補手生成

条件			ショットの種類								
コーナーガードがあるかどうか？	リンク内に相手のストーンがあるかどうか？	リンク内に自分のストーンがあるかどうか？	コーナーガード (4種類)	カムアラウンド (5種類)	フリーズ※ (3種類)	ティーライン上にドローショット※	タップショット	ヒットアンドステイ	ランバック	ガードショット	スプリット
×	×	×	○								
×	×	○	○							○	○
×	○	×	○	○	○	○		○	○		
×	○	○	○	○	○	○		○	○	○	○
○	×	×		○				○	○		
○	○	×		○	○	○		○	○		
○	×	○		○			○			○	○
○	○	○		○	○	○	○	○	○	○	○

表 B. 8 エンドプラン Score 3 or more の候補手生成

条件			ショットの種類							
コーナーガードがあるかどうか？	リンク内に相手のストーンがあるかどうか？	リンク内に自分のストーンがあるかどうか？	コーナーガード (4種類)	カムアラウンド (5種類)	フリーズ (3種類)	タップショット	ヒットアンドステイ	ランバック	ガードショット (2種類)	
×	×	×	○							
×	×	○	○	○			○			○
×	○	×	○	○	○	○	○	○	○	○
×	○	○	○	○	○	○	○	○	○	○
○	×	×								
○	○	×			○	○	○	○		
○	×	○		○		○	○			○
○	○	○		○	○	○	○	○	○	○

表 B.9 の※はフリーガードゾーンルールで相手のストーンを排除できない場合に候補手に加えることを指す。

表 B. 9 エンドプラン Prevent Steal の候補手生成

条件	ショットの種類		
リンク内に相手のストーンがあるかどうか？	ヒットアンドステイ	ランバック	パスショット
×			○
○	○	○	○※

デジタルカーリング AI における 戦略書に基づく序盤定石の構築

森 健太郎 伊藤 毅志

電気通信大学 情報理工学部 情報・通信工学科

デジタルカーリング AI では探索的手法が用いられているが、序盤に精度よく探索を行うことが難しい。そこで本研究では、戦略書に基づきデジタルカーリング AI における序盤定石の構築を行う。序盤定石を既存 AI に導入し、対戦によって有効性を検討する。

Construction of Opening Book in Digital Curling AI from a Book of Curling Strategy

Kentaro Mori Takeshi Ito

University of Electro-Communication

We construct an opening book in Digital Curling AI based on a book of curling strategy. We introduced opening book into AI based on Monte-Carlo tree search and evaluated by competition.

1 はじめに

「デジタルカーリング」は、カーリングの戦略をゲーム情報学の分野から議論するためのシステムであり、AI 同士の大会も開かれている[1][2]。

デジタルカーリングは投げたストーンの初速ベクトルに乱数が加わる不確定ゲームである。このことから、大会上位プログラムでは乱数を用いるモンテカルロ木探索のような手法が用いられている。この手法の問題として、終局までの手数が多い序盤では精度よく探索を行うことが難しい点が挙げられる。

このような問題に対して将棋やチェスのようなゲーム AI では戦略の知識に基づく序盤データベースの導入が行われている。定石データベースを導入する利点として「探索よりも有効な手を選択できること」「制限時間のある対局での思考時間の節約」が挙げられる[3]。現状では、戦略の知識に基づく序盤データベースを導入した AI は存在しない。

そこで本研究では、カーリング戦略書に基づき、

デジタルカーリングにおける序盤定石を構築する。既存手法の AI に序盤定石を導入し、対戦によってその有効性を検証する。

2 カーリング

2.1 ルール

カーリングは、氷上の約 40 メートル先の「ハウス」と呼ばれる円をめがけて、2 つのチームが交互にストーンを滑らせるスポーツである。各チーム 8 投ずつ、合計 16 投を 1 エンドとして、通常 10 エンドの合計得点を競う。

得点計算は 16 投終了時点で行われ、ハウス内にあるストーンのうち、ハウスの中心に最も近いストーンを持つチームが得点権を得る。相手のストーンよりハウスの中心に近いストーンの数 が得点となる。1 エンド目の先手後手はコイントスなどで決め、2 エンド目以降は前のエンドで得点したチームが先手となる。どちらのチームも得点しなかった場合を「ブランクエンド」と呼び、先手後手を入れ替えずに次のエンドを行う。

序盤における特殊なルールとして、フリーガー

ドゾーンルールと呼ばれるものがある。各チーム 2 投ずつ、合計 4 投が投げ終わるまで、フリーガードゾーン (図 1 の緑のゾーン) にある相手のストーンをプレイエリアから弾き出すことはできない。

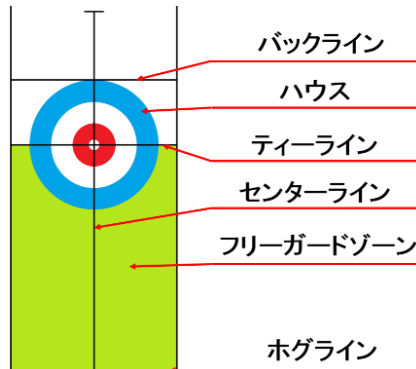


図 1: カーリングのリンク

2.2 戦略

この説では、カーリングの戦略書である「Introduction to Curling Strategy」[4] (以下、戦略書) に基づき、カーリングにおいて重要な戦略の要素について述べる。

2.2.1 エンドプラン

カーリングでは 10 エンドの合計得点を競う。そのため、各エンドをプレイするにあたって現在のエンドをどのように終えるかというエンドプランを設定する必要がある。戦略書ではエンドプランを、このエンドの目標得点と、どの程度リスクを冒して目標得点を取りに行くかという積極性の組で表現している。エンドプランは主に現在のエンド数や得点差・自チームの先手後手・などを考慮してエンド開始時に決定される。

2.2.2 フリーガードゾーンルール

フリーガードゾーンルールの存在により、序盤に置いた石は残りやすく、中盤以降のストーンの数や位置は序盤の影響を強く受ける。ストーンの数や位置は、得点・失点のしやすさに直結することから、エンドプランを達成しゲームに勝つためには、序盤のショット選択が非常に重要である。

2.2.3 ストーンの位置

各エンドにおいて後手は、最後の 1 投をハウスの中心に投げることで確実に得点することが可能である。そのため、後手はセンターライン付近

の相手のストーンを排除しつつ、リンクのサイドでプレイすることが有効である。一方で先手は、後手の得点を防ぐために、積極手にセンターライン付近にストーンを配置するのが有効とある。

また、ハウス内ではティーライン付近にストーンを配置するのが良いとされる。これは、ハウスの奥や手前にあるストーンは相手に利用される危険性が高くなるためである。

3 関連研究

デジタルカーリング AI の強化を目的として、将棋や囲碁の AI で用いられている手法を導入する試みが行われている。囲碁において結果を残しているモンテカルロ木探索を導入したデジタルカーリング AI として、第 1 回 UEC 杯デジタルカーリング大会優勝プログラムの「歩」があり、そのソースコードが公開されている[5]。

このプログラムではランダムシミュレーション中にある程度ストーンの配置が近い局面は同一局面とみなし、シミュレーション回数が増加に伴い局面認識の精度を上げる手法を用いることで、デジタルカーリングにおける局面の連続性に対応している。モンテカルロ木探索の問題点として、1 プレイアウト当たりのシミュレーション回数が多い序盤では、十分なプレイアウト回数が稼げない点が挙げられる。そのため、「歩」では作者のヒューリスティックによる候補手の絞り込みや 1 投目での決め打ちなどを行っているが、体系的な戦略の知識に基づいたものではないため、その妥当性には疑問の余地がある。

本研究では、戦略書に基づく序盤定石を導入することで、持ち時間の節約による中盤以降のモンテカルロ木探索の精度向上を試みるとともに、序盤では探索によって得られるショットより良いショットを選択することで、AI の強化を図る。

4 提案手法

4.1 概要

提案手法では、AI の強化を目的として、戦略書に基づく序盤定石を構築した。定石により、探索よりも有効なショット選択の実現を試みる。また、大会のレギュレーションでは 1 ゲームあたりの持ち時間内であれば、各手番に割り振る思考時間は自由であるため、序盤の思考時間の節約によって、中盤以降の探索にかけられる時間の増加すること期待できる。

提案手法によるショット選択手順を図 2 に示す。

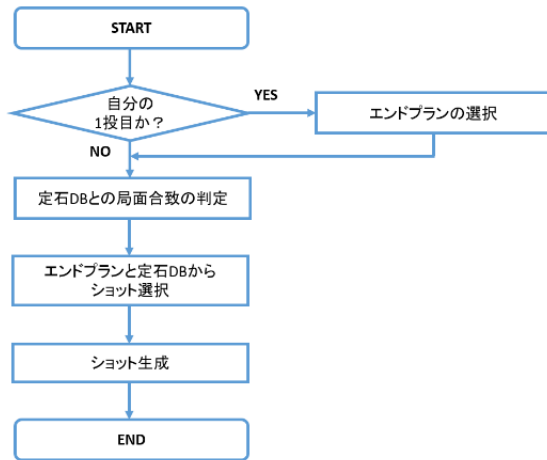


図 2: 提案手法によるショット選択手順

4.2 エンドプランの選択

エンドの開始時にエンドプランを選択する。エンドプランの選択は現在のエンド数と得点差・先後から、戦略書に基づいて作成したテーブルによって、目標点数と積極性を選択することで行う。積極性は、0, 1, 2, 3 の離散値で表し、数値が大きいほど積極性も大きくなるとする。

4.3 局面合致の判定

ストーン的位置が十数値で表されるとき、カーリングにおける局面の数は無数に存在する。そのため、現在局面と定石 DB に登録された局面との合致を判定するために、ストーン的位置を離散化する必要がある。

そこで提案手法では、シートを分割し、ストーンがどのエリアに存在するかによって局面の合致を判定する。分割は図 3 のように 13 のエリア（ハウス 9 分割、フリーガードゾーン 3 分割、その他 1 つ）に分けることで行う。これは、2.2 節で述べたように、序盤においてはハウス内の前後左右、フリーガードゾーンの左右どこにストーンが存在するかが重要なためである。

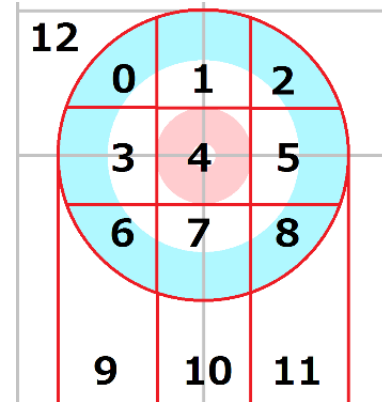


図 3: 提案手法によるリンク分割

4.4 定石 DB に基づくショット選択

カーリングでは同一曲面であってもエンドプランによってショット選択は異なる。そこで、戦略書もとに、様々な曲面についてエンドプランに応じたショットを記述し、定石 DB として構築した。提案手法では定石 DB に基づいてショットの選択を行う。

図 3 に 1 投目の先手のショットに対する、2 投目の後手のショット選択の例を示す。後手の目標得点が 2 であるとき、現在局面に対する、積極性に応じたショットの選択肢は図下段に示す 3 つとなる。積極性が 3 の場合は下段右のショットが選択される。

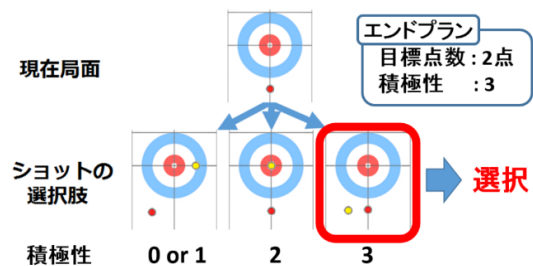


図 3: 定石 DB によるショット選択

4.5 ショット生成

3.4 節で選択したショットに対して、実際にデジタルカーリングの対戦クライアントへ送信するショットベクトルの生成を行う。ほかのストーンに干渉しないようなショットに関しては、あらかじめ計算しておいたショットベクトルを返す。ストーンははじき出すショットなど、ほかのストーンに干渉するショットについては、デジタルカーリングの公式シミュレータで提供されている関数を用いて生成したショットベクトルを返す。

5 実験

5.1 概要

提案手法の有効性を検証するために、提案手法を「歩」に組み込んだもの（以下、提案システム）を用意した。提案システムのショット選択手順を図 5 に示す。

4.2 節では、序盤のショット選択を議論する前提として、序盤のショット選択がゲームの勝率に影響するかどうかの検証実験を行った。4.3 節では、提案手法の有効性を評価するために、提案システムと提案手法を組み込む前の「歩」（以下、既存システム）の対戦実験を行った。

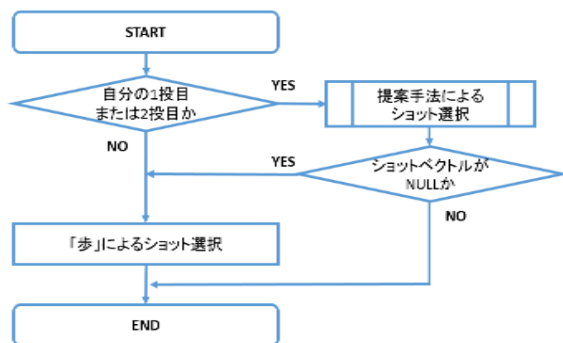


図 5: 提案システムによるショット選択手順

5.2 予備実験

5.2.1 目的

本実験では、デジタルカーリングにおいて序盤のショット選択がゲームの勝率に影響するかを検証するために、提案システムと序盤を考えないプログラムとの対戦を行った。

5.2.2 方法

序盤を考えないプログラムとして、各エンドで最初の 2 投をスルー（ほかのどのストーンにも干渉せず、プレイエリア外に出るショット）し、3 投目以降を「歩」で思考するプログラム（以下、Through）を用意した。それ以外の条件を揃えるために、提案システム、Through とともに、探索による思考時間は 1 手 5 秒に固定し、先手後手 150 試合ずつ、合計 300 試合の対戦を行った。試合設定は以下に示すとおりである。

- 乱数の大きさ：0.145
- 持ち時間：438 秒
- エンド数：10 エンド

5.2.3 結果

表 1: 序盤を考えないプログラムとの対戦結果

提案システムの勝利	引き分け	勝率	p
165	45	0.717	0.000

5.2.4 考察

表 1 の結果より、提案システムが有意水準 5% で有意に勝ち越したことから、デジタルカーリングは序盤のショット選択が勝率に影響するゲームであるといえる。この結果を踏まえ、4.3 節では定石によるショット選択の有効性について検討する。

5.3 提案手法の評価実験

5.3.1 目的

本実験では、提案手法の有効性を検証するために、提案システムと、提案手法を組み込む前の「歩」（以下、既存システム）との対戦実験を行った。

5.3.2 方法

• UEC 杯と同一条件での対戦

AI の強さの向上を確認するために UEC 杯デジタルカーリング大会と同一条件での先手後手 250 試合ずつ、合計 500 試合の対戦を行った。また、持ち時間の節約による中盤以降の探索時間の向上を確認するために、同様の条件で 1 手あたりの平均思考時間と平均プレイアウト数の比較を行った。UEC 杯の試合設定を以下に示す。

- 乱数の大きさ：0.145
- 持ち時間：438 秒
- エンド数：10 エンド
- 思考を固定した場合の対戦

定石による序盤のショット選択と、モンテカルロ木探索による序盤のショット選択のどちらが有効であるか検証する対戦実験を行った。定石による持ち時間の節約の影響を排除するために、探索による 1 手あたりの思考時間を 5 秒に固定した。それ以外は上記の UEC 杯と同一条件で、先後 250 試合ずつ、合計 500 試合の対戦を行った。

5.3.3 結果

表 1 に UEC 杯と同一条件での対戦結果、および表 2 に探索による 1 手あたりの思考時間を固定した場合の対戦結果を示す。勝率は引き分けを除いて計算し、片側二項検定による p 値を算出した。

表 2 に提案システムと既存システムそれぞれ

の 1 手あたり平均思考時間と平均プレイアウト数を示す。

表 2: UEC 杯と同一条件での対戦結果

提案システムの勝利	引き分け	勝率	p
227	87	0.550	0.024

表 3: 思考時間を固定した場合の対戦結果

提案システムの勝利	引き分け	勝率	p
203	105	0.514	0.307

表 4 思考時間とプレイアウト数の比較

	提案システム	既存システム
思考時間 [ms]	6451.9	5816.3
プレイアウト数 [回]	1.90×10^6	1.66×10^6

5.3.4 考察

表 1 より, UEC 杯の条件下では有意水準 5% で提案システムが有意に勝ち越した. このことから, 提案手法を既存の AI へ組み込むことで, 強さの向上を図れることが確認できた. また, 表 2 の結果から, 1 手あたりの思考時間とプレイアウト数が向上していることから, 提案手法によって期待される効果である「序盤の思考時間の節約による中盤以降の探索にかけられる時間の増加」が確認できた.

一方で 1 手あたりの思考時間を固定した場合の対戦では, 勝率に有意差はみられなかったことから, 「探索よりも有効なショット選択の実現」は確認出来なかった. しかしながら, 勝率の低下はみられなかったことから, 少なくとも探索によって得られるショットと同等のショット選択を, 十分に短い時間で行うことができることが確認できた.

思考時間を固定した場合の対戦ログから実際に行われたショット選択を調べたところ, 序盤では定石により局面をシンプルにするようなショットを選択しているにもかかわらず, 中盤以降では探索によって局面を複雑にするようなショットを選択している試合がみられた. これは中盤以降の探索にはエンドプランの導入を行っておらず, 序盤と中盤以降で戦略を共有できていないことによるものである. このような定石による序盤の戦略と探索による中盤以降の戦略の不一致が, 強さが有意に向上しなかった原因ではないかと考えられる.

6 おわりに

本研究では, デジタルカーリング AI の強化を目的として, 戦略書に基づく序盤定石の構築を行った. 序盤定石を既存手法 AI に導入することで, UEC 杯の大会環境においては AI を強化できることを確認した. 序盤の思考時間節約による中盤以降の探索時間向上の効果は確認できたが, 定石によるショット選択を中盤以降に生かしていないといった問題点もみられたことから, 中盤以降にも戦略の知識を導入する必要があると考えられる.

本手法を導入した AI である「GCCS」は, 第 2 回 UEC 杯デジタルカーリング大会において優勝を収めた[6].

今後は, 中盤以降にも戦略の知識を導入することで, 本手法の問題点の改善を行う.

参考文献

- [1] 北清勇磨, 岡田雷太, 伊藤毅志: デジタルカーリングサーバの提案と紹介, 情報処理学会ゲーム情報学研究会報告, 2014-GI-31(2), pp.1-5 (2014).
- [2] 伊藤毅志, 森健太郎, 北清勇磨: 第 1 回 UEC 杯デジタルカーリング大会報告, 情報処理学会ゲーム情報学研究会報告, 2015-GI-34 (2), pp.1-5 (2015).
- [3] 増子直樹, 田中哲郎, 定跡データベースの評価法の提案, 情報処理学会ゲーム情報学研究会報告, 2013-GI-30(3), pp.1-8 (2013).
- [4] Gabrielle Coleman, Introduction to Curling Strategy, ISBN 9781941164020 (2015)
- [5] 第 1 回 UEC 杯デジタルカーリング大会, デジタルカーリング, 入手先 (<http://minerva.cs.uec.ac.jp/curling/wiki.cgi?page=%C2%E81%B2%F3UEC%C7%D5%A5%C7%A5%B8%A5%BF%A5%EB%A5%AB%A1%BC%A5%EA%A5%F3%A5%B0%C2%E7%B2%F1>) (参照 2015-01-20)
- [6] 第 2 回 UEC 杯デジタルカーリング大会, デジタルカーリング, 入手先 (<http://minerva.cs.uec.ac.jp/curling/wiki.cgi?page=%C2%E82%B2%F3UEC%C7%D5%A5%C7%A5%B8%A5%BF%A5%EB%A5%AB%A1%BC%A5%EA%A5%F3%A5%B0%C2%E7%B2%F1>) (参照 2016-01-20)

デジタルカーリング AI 「じりつくん」

加藤 修 飯塚 博幸 山本 雅人
北海道大学大学院情報科学研究科

じりつくんはデジタルカーリングをプレイする AI である。投球の目標座標と回転方向を候補手、盤面状態をノードとし、不確実性を考慮した Minimax 法によるゲーム木探索を行う。

Digital Curling AI "Jiritsu-kun"

SHU KATO HIROYUKI IIZUKA MASAHIITO YAMAMOTO
Graduate School of Information Science and Technology

Jiritsu-kun is AI playing digital curling by minimax game tree search method, where candidate moves are represented as the combination of the throwing-target position and rotational direction of stones and the board state are nodes of the tree.

1. 候補手の生成

デジタルカーリングへの入力は投球するストーンの初速度 v (2次元実数値ベクトル) と回転方向 R (右回転, 左回転の2値) であるが, (v, R) は投球目標座標 q と R の組 (q, R) と一対一で変換可能である。さらにデジタルカーリングでは入力の v に乱数が加えられるが, 結果的に本来の q からのずれが x, y 方向に各々平均 0 標準偏差 σ_x, σ_y (GAT 杯では $\sigma_x = 0.5r_s, \sigma_y = 2r_s$ (r_s : ストーン半径)) の正規分布に従うように設計されている¹。そのため乱数による結果の不確実性に対する考慮のしやすさから, じりつくんではゲーム木探索における候補手を (q, R) と定義している。

また, q は実数値であり無数に存在するため, 有限個に絞る必要がある。じりつくんではプレイエリア内 (図 1) にストーンを停止させるドロウと, プレイエリア内のストーンをはじき出すテイクアウトの二種類の候補手を生成するために, 有限個の要素をもつ投球目標座標集合 Q_d と Q_t を生成する。 Q_d はドロウを想定してプレイエリア内の一部 (図 2 の点

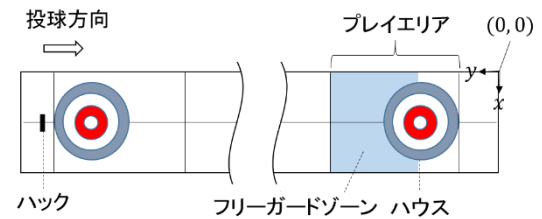


図 1 カーリングリンク

線の枠①) を x 方向に $0.5r_s$, y 方向に $2r_s$ で離散化した投球目標座標集合であり, Q_t は初速度の大きい候補手を生成するためにプレイエリア遠方の座標を横一列 (図 2 の点線②) に $0.5r_s$ の間隔で離散化した投球目標座標集合である。 Q_d の要素数は 1581 個, Q_t の要素数は 83 個であり, 各投球目標座標から右回転, 左回転の二種類の候補手を生成するため, じりつくんにおける候補手数は合計で 3328 個となる。

2. 不確実性を考慮したゲーム木の生成

じりつくんのゲーム木探索におけるノードとは, ある候補手による投球シミュレーショ

¹ 厳密にはハウス中央以外を投球目標座標とした場合はこの分布には従わないが, ここでは無視する。

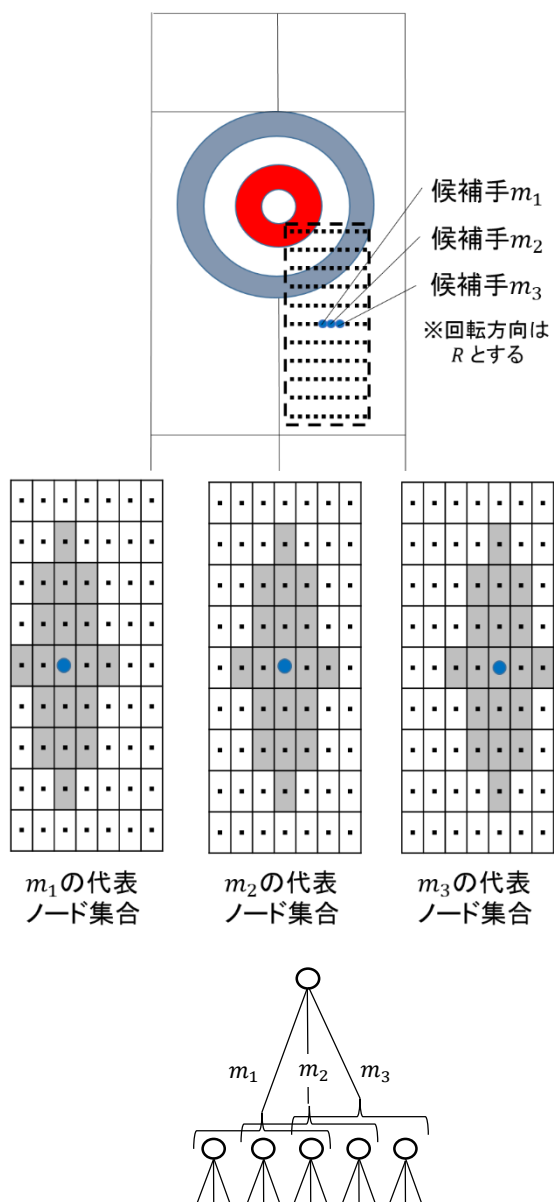


図 5 候補手間での代表ノードの重複

3. 局面評価関数

局面評価関数とは局面 s の善し悪しを評価する関数であり，じりつくんでは以下の式(1)で表される関数を用いている．

$$e(s) = W(s) + \mu \sum_i u(s, i) N(s, i) \quad (1)$$

$W(s)$ は局面 s でエンドを打ち切り，次のエンドに進んだ場合の勝率を出力する関数である．具体的には，局面 s のストーン配置で本

表 1 勝率テーブル（一部抜粋）

先攻	残りエンド数					
	3	2	1	0		
得点差	-3	5%	3%	1%	0%	
	-2	11%	8%	4%	0%	
	-1	20%	17%	15%	0%	
	±0	37%	36%	31%	31%	
	+1	58%	61%	54%	100%	
	+2	76%	80%	85%	100%	
	+3	88%	91%	97%	100%	
後攻	残りエンド数					
	3	2	1	0		
得点差	-3	12%	9%	3%	0%	
	-2	24%	20%	15%	0%	
	-1	42%	39%	46%	0%	
	±0	63%	64%	69%	69%	
	+1	80%	83%	85%	100%	
	+2	89%	92%	96%	100%	
	+3	95%	97%	99%	100%	

見方の例

「後攻，残りエンド数 2，得点差 0」の状況で 1 点取った場合，次のエンド開始時の試合状況は「先攻，残りエンド数 1，得点差+1」であり勝率 54%となる．

来後攻 8 投目投球後に行われる得点計算を行い次のエンドに進んだ場合の「残りエンド数」，「現在の得点差（自分の得点 - 相手の得点）」，「自分の手番（先攻 or 後攻）」の 3 つの要素を確定させ，表 1 の勝率テーブルを参照する（勝率テーブルの作成方法は省略する）．ただし局面 s が先攻投球後の局面である場合には，先攻のストーンの中でハウス中央に最も近いストーン（先攻のナンバーワンストーン）を後攻のストーンに置き換えて得点計算を行う．これは本来の得点計算は必ず後

攻投球後に行われるため、先攻のナンバーワンストーンは後攻の投球によりはじき出されやすいという点を考慮している。

また、 $\mu \sum_i u(s,i)N(s,i)$ は局面 s のストーン配置を評価する項であり、 μ は結合の重み、 $u(s,i)$ は局面 s における i 番目に投球されたストーン $b(s,i)$ が自分のストーンである場合は +1、相手のストーンである場合は -1 を返す関数である。 $N(s,i)$ は $b(s,i)$ の位置、他のストーンとの位置関係の評価する関数であり、以下の式(2)で表される。

$$N(s,i) = J(s,i) \times T(s,i) \times D(s,i) \quad (2)$$

$J(s,i)$ は $b(s,i)$ の得点への結びつきやすさに関する評価指標、 $T(s,i)$ は $b(s,i)$ のテイクアウトされにくさに関する評価指標、 $D(s,i)$ は $b(s,i)$ のダブルテイクアウトされにくさに関する評価指標を表す。 図6～図9はある状況における $J(s,i)$, $T(s,i)$, $D(s,i)$ の値をヒートマップで表した図である（赤いほど評価値が高い）。 図6は各座標ごとの $b(s,i)$ を置いたときの $J(s,i)$ の値、図7～図9は既に $b(s,i)$ が盤面上にある場合に各座標に別なストーンを置いたときの座標ごとの $T(s,i)$ の値（図7,8）と $D(s,i)$ の値（図9）を表す。

$J(s,i)$, $T(s,i)$, $D(s,i)$ の詳細な式は以降の節で述べる。

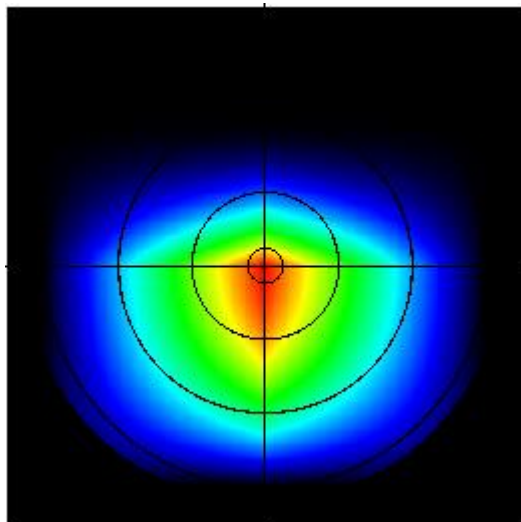


図 6 $J(s,i)$ のヒートマップ

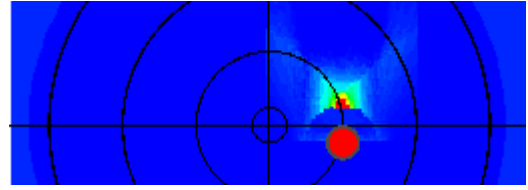


図 7 $T(s,i)$ のヒートマップ 1

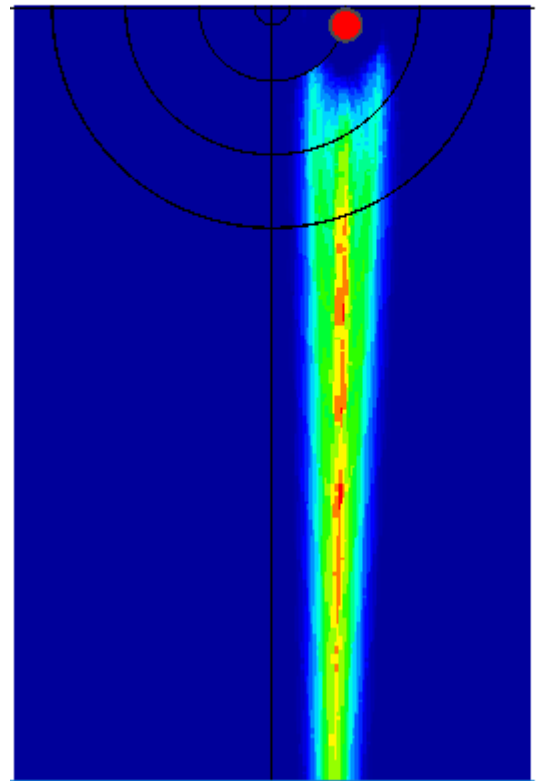


図 8 $T(s,i)$ のヒートマップ 2

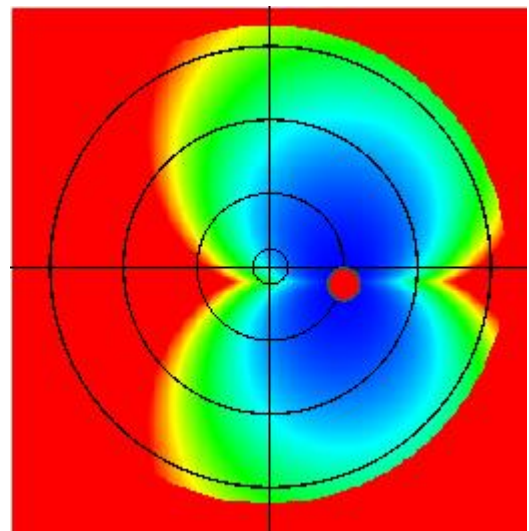


図 9 $D(s,i)$ のヒートマップ

3.1 得点への結びつきやすさ $J(s, i)$

$$J(s, i) = w_1 k(s, i) + w_2 h(s, i) \quad (3)$$

$$k(s, i) = \frac{1}{1 + n_s(s, i)} \quad (4)$$

$$h(s, i) = h_x \times h_y$$

$$h_x = \begin{cases} 1.0 - \frac{|x - C_x|}{r_{h3} + r_s} \\ (|x - C_x| < r_{h3} + r_s) \\ 0 \text{ (otherwise)} \end{cases} \quad (5)$$

$$h_y = \begin{cases} \frac{(y - C_y - r_{h2})^2}{r_{h2}^2} \\ (C_y - r_{h2} < y \leq C_y) \\ 1.0 - \frac{(y - C_y)^2}{r_{h3}^2} \\ (C_y \leq y < C_y + r_{h3}) \\ 0 \text{ (otherwise)} \end{cases} \quad (6)$$

- w_1, w_2 : 結合重み
 $n_s(s, i)$: $b(s, i)$ よりハウス中央に近い
 相手ストーンの数
 (C_x, C_y) : ハウス中央の座標
 r_{h2}, r_{h3} : 図 2 参照
 r_s : ストーン半径0.145

3.2 テイクアウトされにくさ $T(s, i)$

$$T(s, i) = \min(o(s, i, R)) + c \quad (7)$$

$$o(s, i, R) = w_3 o'(o_f(s, i, R)) + w_4 o'(o_b(s, i, R)) \quad (8)$$

$$o'(o_f(s, i, R)) = \frac{1}{o''((o_f(s, i, R)) + 1)} \quad (9)$$

$$o'(o_b(s, i, R)) = \frac{1}{o''((o_b(s, i, R)) + 1)} \quad (10)$$

c : 定数

w_3, w_4 : 結合重み

式 (7) 中の $\min(o(s, i, R))$ は R が右回転の場合と左回転の場合で値の小さい方を返す。

$o(s, i, R)$ は相手が回転 R で投球した場合の $b(s, i)$ のテイクアウトされにくさに関する評価指標であり、この評価指標の計算にはストーン同士の相対位置とストーンのテイクアウトされにくさの関係についての事前実験の結果を用いている。以下に事前実験の具体的な内容を述べる。

ストーン b' をストーン b より投球位置に近い位置に配置し、 b をテイクアウトする軌道で投球する。テイクアウトは b の左端にぎりぎり接触する軌道からはじめて軌道を $r_s/6$ ずつ右にずらして計 24 軌道行う。結果はテイクアウトの軌道ごとに、投球したストーン b_t が b に接触した場合はテイクアウト成功で 1、接触しなかった場合はテイクアウト失敗で 0 として 24bit の bit 列で保存する。これを b_t の回転を右回転、左回転の 2 通りで、さらに b の位置を固定して b' 位置を少しずつ変えて行い、 b と b' の相対位置ごとにテイクアウトの成功と失敗に関する 24bit の bit 列のデータベース O_f を作成した。同様のことを、 b' を b より投球位置から遠い位置に配置して行い、データベース O_b を作成した。ただしこの場合には、 b_t は 24 軌道すべての軌道で b と必ず接触するため、テイクアウトの成功条件は b が元の位置から一定以上移動することとした。

$o_f(s, i)$, $o_b(s, i)$ は上記の方法で作成した O_f, O_b を参照する関数である。 $b(s, i)$ と局面 s における $b(s, i)$ 以外のすべてのストーンとの相対位置ごとに bit 列を参照し、それらを OR 演算した結果を出力する。また、 o'' は

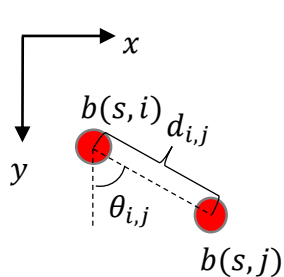


図 10 式 (12) (13) 中の $d_{i,j}, \theta_{i,j}$

bit 列を入力とし, bit 列中で 1 が最も多く続いた回数を出力する.

3.3 ダブルテイクアウトされにくさ $D(s,i)$

$$D(s,i) = \min(D'_{i,j}) \quad (11)$$

$$D'_{i,j} = \begin{cases} 0.5 + 0.5 \frac{d_{i,j}^2}{D''_{i,j}} & (d_{i,j} < D''_{i,j}) \\ 1 & (otherwise) \end{cases} \quad (12)$$

$$D''_{i,j} = a + b \cos \theta_{i,j} \quad (13)$$

a, b : パラメータ

j : i 以外の自分の石の番号

$d_{i,j}, \theta_{i,j}$: 図 10 参照

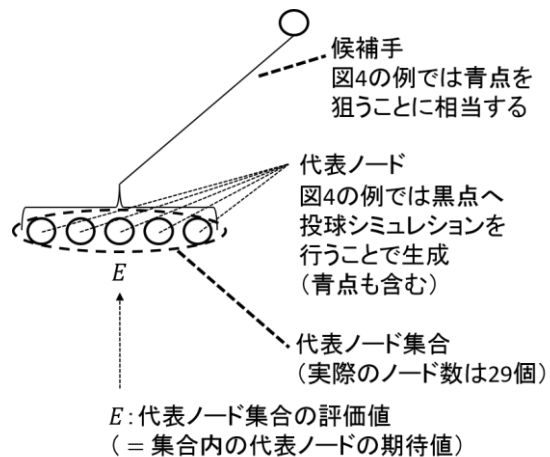


図 11 ゲーム木探索 1

式 (11) 中の $\min(D'_{i,j})$ はすべての j のうち $D'_{i,j}$ の値が最小となる場合の $D'_{i,j}$ の値を返す.

4. まとめ

じりつくくんは現在深さ 2 までの探索を行うが, 探索の全体像を図 11 と図 12 に示す.

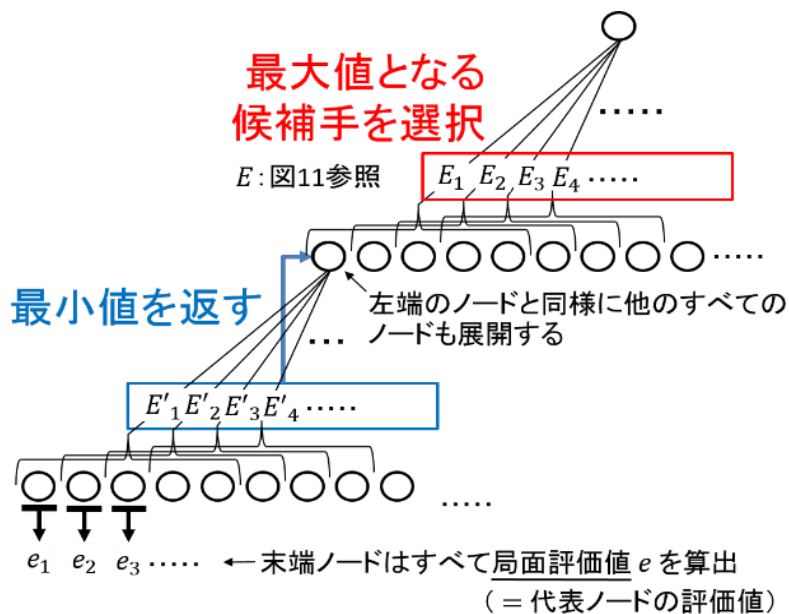


図 12 ゲーム木探索 2

思考ゲーム AI 制作を行う授業の取り組み

金井 誠¹ 岡田 拓也² 樋口 利幸² 宮下 優希² 伊藤 毅志²

1) 電気通信大学院 情報理工学専攻 2) 電気通信大学 情報理工学部

概要 近年、人工知能への関心が広く高まっており、大学などの学術的機関においても人工知能に関する教育の需要は高まっていると考えられる。そこで本研究室では 5 五将棋を題材とした、思考ゲーム AI プログラムを作成する授業を行っている。本論文ではこの授業の流れについて説明し、本年度の受講生各々の取り組みや今後の課題点について述べる。

1. はじめに

近年、人工知能への関心が広く高まっている。この関心は学術的関心に留まらず、社会全体を巻き込んだ大きなムーブメントとして幅広い層の人々の注目を集めている。ゆえに、大学などの学術的機関においても人工知能に関する教育の需要は高まっていると考えられ、そういった普遍的かつ初学者向けの人工知能分野の学習が可能な題材が求められている。

そこで本研究室では 2012 年から授業の 1 つとして、5 五将棋やどうぶつしょうぎを題材とした、思考ゲーム AI プログラムを作成する授業を行っている。本論文ではこの授業の内容と流れについて説明し、本年度の受講生各々の取り組みや今後の課題点について述べる。

2. 情報工学工房について

本授業は電気通信大学情報理工学部情報・通信工学科で開講されている「情報工学工房」という授業で行われている [1]。この授業は全学年で開講されており、1 年を通して授業が行われる他、強く関心を持つ学生に対しては 2 回以上の受講も認めている。授業を受講する学生は幾つか出されるテーマから好きなものを選ぶことができ、各テーマ毎にグループに分かれてテーマに沿った演習を行う。そして、年度末に 1 年を通して学習したことを授業全体の発表会で各コース毎に発表して評価する。

このうち、本研究室が担当しているテーマは「思考ゲーム AI を作ろう」である。ここでは将棋と囲碁を題材に AI を作成し、現在のゲーム AI がどのようにして作られているのかを学習することを目的としている。この内、本研究室は将棋 AI を担当しており、囲碁は他の研究室の担当となっている。

3. 授業内容

3.1 授業スケジュール

まず始めの 5 ～ 6 月の間は題材となるゲームを実際に遊び、ゲームがどのような性質を持っているのか、どういった戦略が有効になるのかを実践から学習させた。ここでは将棋、どうぶつしょうぎ、囲碁について遊んでもらい、将棋と囲碁の 2 つの間のゲームとしての違いなどについても学習してもらった。

その後、7 月から希望するコースに別れ、各々プログラムの作成の演習へ移行した。本研究室が担当した将棋コースの授業では、プログラムの構造を理解してもらうために、まずどうぶつしょうぎのプログラムを作成してもらい、夏休み明けから 5 五将棋のプログラムの開発を行う流れで進めた。そして、2 ヶ月間将棋 AI の基礎知識を学習してもらい、11 月に電気通信大学で開催される UEC 杯 5 五将棋大会 [2] に参加してもらった。

3.2 授業内で取り組んだ将棋 AI の基礎理論

本授業では主に将棋 AI の基礎理論から説明を行っており、昨今の将棋 AI で広く用いられている技術の中でも、実装面でやや高度になるものは本講義では教えていない。これは授業の受講生の知識レベルに幅があるため、将棋 AI を実装するためには基本的な知識から教える必要があるためである。講義に割ける時間も限られているため、昨今用いられている技術の全てを教えるには時間も足らず、限られた時間の中でより有効であると考えられる技術に焦点を当てて教えている。

しかしながら完全に 1 からプログラムを作成する場合、画面への盤面の出力やコマンド入力の受付部分、入力コマンドのフォーマット一致判定等、プログラム中のゲーム AI たらしめる部分以外にも制作に割く必要がある。これらの実装は比較的容易ではあるが、不本意なバグなどが出ることを考慮すると、これらの実装で時間を取られ

てしまうことはなるべく避けたい。そこで、この授業では基本的な入出力機能や、合法手生成、合法手判定などの基本機能の実装の肩代わりを担うライブラリを受講生に配布し、実装上の負担を減らしている。ライブラリでは盤面操作や合法手生成のアルゴリズム、 $\alpha\beta$ 探索の概形については実装済みであり、これらの実装については実際に受講生は殆ど行っていないものの、昨今の将棋 AI の技術的基礎基盤となっているため、授業内でも講義の形で教えている。その為、授業内の演習はライブラリでは実装されていない、ゾブリストハッシュやトランスポジションテーブルのような機能追加や、反復深化を組み合わせた Move Ordering の実装などの探索の改良が主体になっている。

3.3 個人の興味に沿った改良案

授業内容としては前節に述べたとおり、基礎的なものが殆どで高度な技術については紹介にとどめているが、その後の改良の過程では各々興味のある改良を行っている。ここでは、本年度受講生の各々の取り組みについて紹介する。

まず、樋口は「強い将棋の AI」の実現を目指し、前述の機能の実装や細かな改良に力を入れた。王手延長などの延長の長さなどにも制限を加え、細かな調整を加える事で AI の強化を行っている。今後はヒューリスティックな知識を入れることで評価関数の改良を行っていく予定である。

次に、岡田は「深く読む AI」をテーマに、ある程度の強さは保持しつつも強さよりも深く読むことを念頭においた AI を実現することを目指した。王手延長や捕獲探索による探索の延長をなるべく深く行うようにしたり、候補手の絞込にビームサーチを導入することで、良い手を取りつつも計算量を抑え、なるべく深く探索することが出来ないか検討した。これらの実装については現在も検討中であり、今後の課題点の 1 つとなっている。

最後に、宮下は最終的な目標として「強化学習で評価関数を学習する AI」を掲げ、様々な取り組みを行った。その一環として、本研究室で提供した C 言語のライブラリと Node.js を用いて AI をサーバー上で動作できるようにし、ブラウザ上で対戦が出来るようにした。これにより、AI との対戦をより簡単なものにし、様々な人にプレイしてもらえるようにすることで、改良点のテストや棋譜の収集等をやりやすくしたり、改良前のプログラムと改良後のプログラムをサーバー上で実行させ、サーバー上での自動対戦を可能にした。具体的な強化学習への取り組みについては検討項目が多く、まだ実行に移せていないものの、今後機械学習の手法を取り組む上での基礎基盤を気づくことが出来た。

4. おわりに

本論文では本年度の思考ゲームの AI を実装する授業で見られた、受講生各々の取り組みについて纏めた。本年度の受講生はとりわけ個性があり、各個人で特色の持った AI を実装しようという取り組みが見られた。残念ながら当初の各々の目標に到達することは出来なかったが、現在の将棋 AI に用いられている技術の一端を学習できたと考えられる。また、本年度の受講生は 2015 年 11 月 22 日に行われた、第 9 回 UEC 杯 5 五将棋大会 [2] に出場し、他の出場者に対しても勝利するなどの健闘を見せた。

しかしながら、授業としては幾つか課題点も見られ、中でも問題になるのが「受講生各々のやりたいことを実現するための準備」であると考えられる。本年度は宮下が強化学習への興味を示し、それを目指して AI の改良を行ったが実現にまでは至らなかった。幅広い人工知能への関心から、こういった機械学習の手法を学ぶためのテストベッドとしての需要は一定数あると考えられ、今後も機械学習を行う事を目指して本授業を取る学生は出てくるものと思われる。今後はこういった機械学習などの発展的な内容について学びたい学生についても考慮し、幅広い学生に対して適切なサポートを行えるような講義資料・開発環境の整備を行ってきたい。

参考文献

- [1] 電気通信大学：シラバス 情報工学工房 (online), http://kyoumu.office.uec.ac.jp/syllabus/\2015/21/21_17223303.html (参照 2016-2-16).
- [2] 電気通信大学 伊藤毅志研究室：第 9 回 UEC 杯 5 五将棋大会 - 5 五将棋 portal(online), <http://minerva.cs.uec.ac.jp/~uec55shogi/wiki.cgi?\\page=\\%C2%E8%A3%B9%B2\F3UEC\C7\D5%A3\\B5%B8\DE\BE\AD%B4\FD\C2\E7%B2\F1> (参照 2016-2-18).