

人狼ゲームにおける人工知能が学習困難な戦術の研究

飛田 伊織 (指導教員 西村 俊二)

令和2年1月27日

Learning Disability of Artificial Intelligence on Werewolf Game Strategies

IORI HIDA (ACADEMIC ADVISOR SHUNJI NISHIMURA)

概要: 人工知能技術は、囲碁、将棋、チェスなどの「完全情報ゲーム」においては研究が進められており、すでに人工知能が人間に勝利した例も存在する。このことから、人工知能は特定の分野においては人間を凌駕しているといえる。そこで次の研究題材として、「不完全情報ゲーム」である人狼ゲームをプレイする人工知能である「人狼知能」が注目されている。毎年人狼知能エージェントの強さを競い合う大会が開催されるなど、人狼知能の研究は進められているが、ゲームにおけるエージェントの行う戦術と人間の行う戦術には大きなギャップがある。そこで本研究では、人間がよく行う行動の一つとして村人陣営が敗北濃厚時に行う戦術、例としてパワープレイの阻止を組み込んで実験を行い、勝率や動きの変化を観察して、人工知能が学習困難な事項を明らかにしようとした。結果的に今回想定した動きをさせることには成功したものの、勝率の変化は予想と大きく異なっていた。しかし、状況を限定することでパワープレイの阻止が村人陣営の勝率向上に寄与していることがわかり、パワープレイの起こりうる状況の少なさから一部ではあるものの学習が困難である事項を確認することができた。

キーワード: 人工知能, 人狼ゲーム

1. 緒言

近年、第3次AIブームと言われるほど人工知能の関心が高まっており、技術の進展も著しい。特に人間と人工知能を対戦させ、人工知能がどの程度賢くなったのかを理解する動きが活発であり、囲碁や将棋において、人工知能がトッププロを破るという快挙が成し遂げられた。このように、人工知能の技術は分野によっては人間の能力を凌駕している。しかし、プレイヤー間の情報がすべて開示される確定情報ゲームに対し、ゲームに関する情報が隠ぺいされる不確定情報ゲームにおいては、人工知能が人間を負かした例は多くない。また、現在の人工知能は、他者や社会などの周囲の状況を認識して行動を起こすものではなく、そのための技術も確立されていない。現在、社会からのコミュニケーションAIに対する信頼は高くなく、人工知能技術が社会の中で活用されていくには、社会的なコミュニケーションの実現が必要とされる。人狼知能の実現には、「だます」、「見破る」、「説得する」などのコミュニケーションが行えるようになることが必要である。このような理由から、新たな人工知能のジャンルである不確定情報ゲームとして現在注目されているのが人狼ゲームである。

本研究では、不確定情報ゲームの代表として人狼ゲームに焦点を当てる。人狼ゲームを人工知能にプレイさせる動きは既に存在するが、未だ人間に勝利できる人工知能が公になっていない理由として、人間が使っている高度な戦術を、人狼ゲームを行う人工知能(今後

は人狼知能と表記する)はまだ使用できていないからだと考えられる。

今回は人間のあいだでは高度とされている戦術の一つとして、最終日に村人、裏切り者、人狼の3人が生き残った場合、村人が人狼を騙り勝利を得ようとする戦術を取り上げる。人工知能に村人側が人狼を騙るという機能を組み込み、挙動の変化を観察することで、この機能が最初から組み込まれなかった原因を発見し、延いては人工知能が学習困難な事項を明らかにすることができると思われる。

2. 人狼知能

2.1 人狼ゲーム

まず、人狼ゲームについての説明を行う。ゲームのプレイヤーには役職が割り振られ、役職の持つ能力を利用して勝利を目指す。役職は村人陣営の役職と人狼陣営の役職に分かれており、それぞれの陣営は異なる勝利条件を持っている。村人陣営の勝利条件は、人狼を全員追放することである。人狼陣営の勝利条件は、村人の人数を人狼の人数と同じ数まで減らすことである。プレイヤーは与えられたゲームの情報を推察して敵、味方が誰なのかを推理し、自分の属する陣営が勝利するために立ち回る。

村人陣営は与えられた情報やゲームの状況、情報をもとにした対話から、誰が人狼であるか推理を行う。後述する占い師などの役職を持った村人陣営のプレイヤーは特殊な能力があり、ゲームの状況を把握する能

力が村人より高い。それに対して人狼は、ゲーム開始時に仲間の人狼を知ることが出来る。人狼は仲間の人狼と協力しながら人狼であることがばれないように、ときには嘘をつきながら立ち回ることが基本的な戦術となる。人狼は相手を騙すゲームであるとも言われるが、自分の主張を論理的に説明し、周囲の人々を説得するゲームでもある[1]。

2.1.1 ゲームの流れ

ゲームは昼・夜の2つのフェーズで構成され、ゲーム終了まで各フェーズを繰り返していく。

昼のフェーズではプレイヤー全員で議論が行われ、基本的に村人陣営は人狼が誰なのかを探し、人狼陣営は人狼が誰かを悟られないように偽る、後述する村人陣営の特殊な能力を持つ役職は能力によって得られた情報を駆使し、自陣が有利になるように立ち回る。人狼陣営はここでなにもしなければ敗北する可能性が非常に高いため、このフェーズで村人陣営の特殊な能力をもつ役職として偽ることが多い。

昼のフェーズが終了すれば投票に入り、プレイヤーは人狼だと疑わしき人物を追放する。追放されたプレイヤーはゲームから除外され、投票や議論に参加することができない。

夜のフェーズでは、村人陣営の特殊な能力を持つ者はそれを使い(後述の役職説明を参照)、人狼が誰なのかを探す。人狼は人狼以外のプレイヤーを1人選んで襲撃することができる。襲撃されたプレイヤーは死亡者として扱われ、投票で追放されたプレイヤーと同様にゲームから除外される[2]。

2.1.2 各役職について

ここで、今回の研究で使用した役職について説明する。

(1) 村人

村人陣営に属する。他の役職のような能力を持たず、自分の動きではなく他人の動きに注意を払い、考察することが求められる。人工知能ではなく人間が人狼ゲームを行う場合、村人が敢えて占い師などの役職を偽る場合もあるが、今回使用するエージェントには搭載されていないので割愛する。

(2) 占い師

村人陣営に所属する。夜のフェーズにプレイヤー一人を対象にし、その者が人狼であるか人間であるかを知ることができる。ここで重要なのが、後述する人間でありながらも人狼陣営に所属する裏切り者と、通常の人狼を区別することはできない。しかしながら人狼と人間の区別がつくことは村人陣営にとって大きなアドバンテージであるため、村人陣営、人狼陣営の双方ともに特に重要な役職である。

(3) 裏切り者

人狼陣営に所属する。ただし、人狼からは誰が裏

切り者であるか把握されず、裏切り者からも誰が人狼なのかはわからない。それ以外は特殊能力がないただの村人と同様であるが、人狼陣営が勝利した際に勝利となるため、人狼に協力する。勝利人数のカウント時には人間陣営として数えるため、終盤では裏切り者が自ら追放される行動によって勝利することも可能である。

(4) 人狼

人狼陣営に所属する。フェーズ開始時に人狼以外の者を減らすことができる。仲間の人狼同士でフェーズ終了時にコミュニケーションをとることができる。

2.1.3 本研究で使用する戦術

ここでは、通常の戦術と本研究で使用する戦術の差別点を説明する。なお、相違点は村人陣営と人狼陣営の勝敗が決まる最終日のみあるため、それまでの途中の段階は割愛する。状況も限定し、人狼、裏切り者、村人が生き残った場合にのみ検証を行う。

今までの人狼知能に搭載されていたのは、人狼陣営が占い師を騙っている場合、他の2名は偽の占い結果にしたがって投票先を決める、人狼陣営が役職を騙っていない場合はランダムに投票する、裏切り者がパワープレイを用いて人狼陣営を勝利させるかのいずれかであった。パワープレイとは、裏切り者が残っている状況下で人狼が自身は人狼陣営であるとカミングアウトし、裏切り者と協力して票を合わせることで村人陣営の一人を追放する戦術である。

今回検証するのは、村人にパワープレイを阻止させる学習を組み込んだ場合、勝率、会話内容の変化などの挙動の変化を観察する。具体的には、裏切り者が自身は裏切り者であるとカミングアウトしたとき、本物の人狼のカミングアウトと同時に村人も自身が人狼であるとカミングアウトする。そうすることで、裏切り者の目線ではどちらが味方の人狼なのか分からなくなり、村人陣営の勝つ可能性が生まれるというものである。

2.1.4 用いるエージェントについて

今回は毎年開催される人狼知能大会の決勝進出エージェントを用いる[3]。人狼知能大会とは、人狼ゲームを人工知能にプレイさせようと試む人々が集まって行う大会である。人狼知能大会では人狼知能のプロトコルとサーバを公開することで、多くの人がエージェントの作成に参加できるようにしている。人狼知能プロジェクトでは、人狼知能を汎用人工知能の新しい標準問題とすることを目指している[3]。

本研究では、人狼知能大会で公開された人狼知能プラットフォームの中で、決勝に進出したソースの一つである `cn1` を使用し実験を行う[4]。そのエージェントにはパワープレイの戦術がすでに組み込まれていたため、私は村人がパワープレイを阻止する学習機能

を組み込んだ。

役職の内訳は村人が2人、占い師が1人、裏切り者が1人、人狼が1人の計5人である。通常人狼ゲームは10人から15人で行われるのが一般的だが、今回の研究は最終日にのみ焦点をあてるため少人数で問題ないと考えた。

人間同士が人狼をプレイする場合は、会話には自然言語が用いられる。しかし、強い人狼知能の開発を目指しながら同時にゲーム内で自然言語を扱わせることは難しい。そこで表1、表2のような、人狼知能独自のシンプルな会話プロトコルが用いられている。人間が人狼内で行う会話の半数以上は自分の思考を表現するものであり、それが表1、表2の内容にあたる。残りの20%は理由説明、15%が雑談、残りの15%がその他の発言である。また、本研究で使用する各役職を表す単語は表3のとおりである。

表1 プレイヤーの行動に関するプロトコル[1]

対象指定メソッド	メソッドの内容
vote	投票するプレイヤーを決める
attack	人狼が襲撃するプレイヤーを決める
guard	狩人が防衛するプレイヤーを決める
divine	占い師が占うプレイヤーを決める
会話メソッド	メソッドの内容
Talk	村全体への発話を行う
Whisper	人狼だけに対して発話を行う

表2 会話メソッドで用いるプロトコル[1]

発話可能な内容	詳細
estimate	他プレイヤーの役職の推定
comingout	自分の役職を公言する
divined	占った結果を伝える
inqusted	霊媒した結果を伝える
guarded	護衛したことを伝える
vote	投票したいプレイヤーを伝える
attack	人狼が襲撃したい人に投票する
agree	他プレイヤーの発言に同意する
disagree	他プレイヤーの発言に反対する
over	もう話すことはないとき使用
skip	様子見をしたいとき使用

表3 各役職を表す単語[1]

単語	詳細
VILLAGER	村人
WEREWOLF	人狼
SEER	占い師
POSSESSED	裏切り者

2.1.5 勝率の予想

私の予想では村人、人狼、裏切り者の3人が残った

ゲームに絞ると、村人陣営はパワープレイによって本来負けるゲームに対して勝利する可能性が生まれるため、人狼陣営の勝率が低下し、村人陣営の勝率が上昇すると考えた。もしそうなった場合、この戦術を、人間は行うが人狼知能は行わない、高度な戦術と位置付けることができる。

3. 実験

3.1 エージェントの詳細

本研究では、第3回人狼知能大会決勝進出チームのエージェントである cnd1 を用いて実験を行う。cnd1では基本的に以下の要素を持つこととする。

(1) 村人

人狼候補に投票。1 ターン目以降に投票が集まっているプレイヤーがいる場合、そのプレイヤーに投票する。

(2) 占い師

初日0 ターン目にカミングアウトして占い結果を公開。1日目以降は、人狼候補を占う。

(3) 人狼

村人のふりをする。自分の投票で仲間の人狼が最多得票数でなくなる場合は投票先変更。基本は大衆の意見に従う。裏切り者以外で、人狼陣営に投票が多いプレイヤーを襲撃する。

(4) 裏切り者

占い師のふりをする。初日占い師カミングアウトが他にいた場合、そのプレイヤーが人狼と公開。いない場合は人狼候補でないプレイヤーをランダムに人狼と公開する。

また、今回 cnd1 に搭載されていたパワープレイを実行させるソースコードは以下である。

リスト1. パワープレイプログラム(一部抜粋)

```

1: // 最終日人狼カミングアウト
2: talkTactics.add(new TalkCo(Role.WEREWOLF), 1,
3: Day.on(2));
-----
4: //優先度・日付指定、繰り返し無しでTacticを追加する
5: public void add(T e, int priority, Day day) {
6:     add(e, priority, day, Repeat.ONCE);
7: }
-----
8: //とりあえずカミングアウトする
9: public class TalkCo extends TFAFTalkTactic {
10:     private Role role;
11:     public TalkCo(Role role) {
12:         this.role = role;
13:     }
14: @Override

```

```

15: public ContentBuilder talkImpl(int turn, int skip,
16: int utter, TFAFGameModel model, Game game) {
17:     if (!game.getSelf().hasCO()) {
18:         return new
19: ComingoutContentBuilder(game.getSelf().agent,
20: role);
21:     }
22:     return null;
23: }
24: }
    
```

このエージェントのソース提出前にパワープレイの阻止を行う機能が入っていない理由としては、何らかの原因で正常に機能しないからであると思われる。本研究ではその原因を調査し、理由を考察する。また、ソース内にはリスト1の1行目や、以下のリスト2のような整数値を用いた命令がいくつもあり、このソースが大会に提出される以前から幾度と学習を重ねてきて考えられるため、その点も踏まえて考察する。

リスト2. 整数値が用いられている例(占い師)

```

//初日カミングアウト
talkTactics.add(new TalkCo(Role. SEER), 10700,
Day.on(1));
//白判定時、自分のモデルを信じて黒として伝える。
talkTactics.add(new
TalkDivineWithEvilScore5ver(), 9200, Day.any());
TalkTactics.add(new TalkVoteWolfbySeer5ver(),
8500, Day.any());
    
```

また、本論文の以降では連続実行中の学習を実行学習、ソース提出前の学習を事前学習と表記する。

このエージェントが実行学習を行っているという証拠を以下の図1に示す。

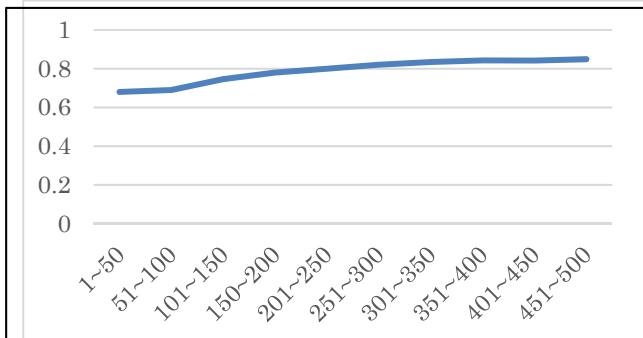


図1. 村人陣営の勝率変化

図1では、ゲームを500回実行したときの村人陣営の勝率変化を表している。1~50回のときは勝率がおよそ68%であり、451~500回のときはおよそ85%と、明らかに変化が生じていることから、このエージェントが何らかの実行学習を行っていると思われる。

3.2 ログの詳細

本研究では勝率のほかにもログにも着目するため、ここで詳細を説明する。例として以下の図2を用いる。

|0,status,5,VILLAGER,ALIVE,DDD↓

図2. ログの例

それぞれの単語が表すのは、左から日付、行の種類、エージェントの番号、エージェントの役職、エージェントの生死、エージェントの名前である。

役職を表す単語については表3で示したので割愛するが、行の種類に関しては表4に示す。

表4 行の種類[1]

単語	詳細
attack	襲撃結果
attackVote	襲撃先投票
divine	占い先指定
execute	追放結果
guard	護衛先指定
result	ゲームの結果
status	各プレイヤーの状態
talk	発話
vote	追放先の投票
whisper	人狼のささやき

3.3 実験の詳細

ここで、本研究で行った実験の詳細を説明する。

- (1) なにも手を加えずに cnd1 のエージェントでゲームを30000回実行し、最終日に人狼、村人、裏切り者が残った場合に限定して双方の勝率を確認する。
- (2) cnd1 のエージェントを村人陣営がパワープレイ阻止をするように改変し、30000回実行して(1)と同じ条件のもと各陣営の勝率を確認する。

その後はログから会話内容や投票先など、挙動の変化を確認する。

なお、本研究の目的は総合的な勝率の変化を確かめることではなく、あくまで「事前学習によってパワープレイ阻止の戦術が組み込まれなかったのはなぜか」を確かめることであるため、状況を「人狼陣営によるパワープレイが起こったとき」のみに限定し、総合的な勝率は集計していない。

3.4 実験結果

3.3の説明に沿って結果を示す。

(1)について

まず、通常のままエージェントを実行した結果、村人陣営と人狼陣営の勝率は以下の表5のように変化した。

た.

表 5. 改変前の実行結果

ゲーム数	562
勝率(村人陣営)	65.3%
勝率(人狼陣営)	34.7%

ここでゲーム数とは、行った 30000 ゲームの中で、最終日に人狼、村人、裏切り者の 3 人のみが残ったゲームの数である。表 5 では 30000 ゲームに対して上記の条件を満たすゲームの割合が低くなっているが、エージェントの構成人数である 5 人の中で人狼は 1 人のみであり、初日に人狼が追放され、早い段階でゲームが終了してしまう場面が多いため、このような数になったと思われる。

パワープレイを行う場面と行わない場面の両方が存在したため、人狼陣営の勝率が極端に高くなることはなかった。パワープレイを必ず行わない理由として、人狼エージェントの目線ではだれが裏切り者か判断する手段に乏しく、カミングアウトできる状況が限られたためと考えられる。

(2)について

エージェント改変後のゲーム数、各陣営の勝率は表 6 のようになった。

ゲームが進むごとに各陣営の勝率がどのように変化したのかについては、表 5, 6 を見ていただければわかる通り、1000 ゲームの実行に対して最終日の条件を満たすパターンがあまりにも少なく、実行学習による勝率の変化は期待できないと判断したため載せないこととした。また、(1)と(2)の条件を満たすゲーム数が表 5 では 562, 表 6 では 417 と大きく異なる点についても、プログラムの改変が原因だと考えられるので、敢えて母数は揃えないこととする。

表 6. 改変後の実行結果

ゲーム数	417
勝率(村人陣営)	39.6%
勝率(人狼陣営)	60.4%

3.4.1 会話の変化

以下の表 7 とリスト 3 に改変を行う前のゲームログの一例と役職配置を示す。また、今後はリスト 3 のゲームをゲーム 1 として表記する。

表 7. ゲーム 1 の役職配置

エージェント 1	VILLAGER
エージェント 2	POSSESSED
エージェント 3	SEER
エージェント 4	VILLAGER
エージェント 5	WEREWOLF

リスト 3. ゲーム 1(改変前)会話

```
2,talk,0,0,2,Skip↓
2,talk,1,0,1,VOTE Agent[05]↓
2,talk,2,0,5,COMINGOUT Agent[05] WEREWOLF
2,talk,3,1,2,Skip↓
2,talk,4,1,5,VOTE Agent[01]↓
2,talk,5,1,1,Skip↓
2,talk,6,2,2,Over↓
2,talk,7,2,1,Skip↓
2,talk,8,2,5,Skip↓
2,talk,9,3,5,Skip↓
2,talk,10,3,1,Over↓
2,talk,11,3,2,Over↓
2,talk,12,4,1,Over↓
2,talk,13,4,5,Over↓
2,talk,14,4,2,Over↓
```

リスト 3 では、2 行目で村人が人狼に対して投票すると宣言を行い、裏切り者の生存を確認した人狼がカミングアウトしてパワープレイを仕掛けている。その後、人狼は自らに投票宣言をした村人に投票宣言し、会話が終了している。

次に、改変後のゲームをゲーム 2 として表記し、ログをリスト 4、役職配置を表 8 に示す。

表 8. ゲーム 2 の役職配置

エージェント 1	VILLAGER
エージェント 2	SEER
エージェント 3	POSSESSED
エージェント 4	WEREWOLF
エージェント 5	VILLAGER

リスト 4. 改変後会話

```
2,talk,1,0,4,VOTE Agent[05]↓
2,talk,2,0,5,COMINGOUT Agent[05] WEREWOLF↓
2,talk,3,1,4,COMINGOUT Agent[04] WEREWOLF↓
2,talk,4,1,5,VOTE Agent[04]↓
2,talk,5,1,3,Skip↓
2,talk,6,2,5,Skip↓
2,talk,7,2,4,Skip↓
2,talk,8,2,3,Over↓
2,talk,9,3,3,Over↓
2,talk,10,3,4,Skip↓
2,talk,11,3,5,Skip↓
2,talk,12,4,3,Over↓
2,talk,13,4,5,Over↓
2,talk,14,4,4,Over↓
```

リスト 4 では、まず 1 行目で人狼が村人に投票すると宣言した後、村人がパワープレイ阻止のためにカミングアウトを行い、直後に人狼もカミングアウトを行っている。その後人狼は村人に対して投票すると宣言を行い、会話が終了した。

3.4.2 特定の条件下での変化

また、改変前にパワープレイが仕掛けられた場合と、改変後にパワープレイの阻止が成立した場合のみの勝率を比較した。結果は表 9 である。

表 9 はゲームを 5000 回実行し、村人、人狼がともに

カミングアウトした場合のみを抽出し勝率を集計している。このような状況のみを見れば、パワープレイ阻止の戦術は村人陣営の勝率向上に寄与しているといえる。

だが実際は改変することで総合的な勝率は予想と逆の変化を及ぼしていた。その理由については次章で考察することとする。

表 9. 戦術成立時の勝率

改変前(村人勝率)	0%
改変後(村人勝率)	50.4%
改変後(人狼勝率)	49.6%

3.5 結果解析

3.5.1 カミングアウトの変化

カミングアウトの変化について、以下に私が注目した2回のゲームログをリスト5, リスト6に示す。

リスト 5. パワープレイ阻止成功例

```
2,talk,0,0,3,VOTE Agent[02]↓
2,talk,1,0,2,COMINGOUT Agent[02] WEREWOLF↓
2,talk,2,0,1,Skip↓
2,talk,3,1,1,Skip↓
2,talk,4,1,2,VOTE Agent[03]↓
2,talk,5,1,3,COMINGOUT Agent[03] WEREWOLF↓
2,talk,6,2,3,Skip↓
2,talk,7,2,1,Over↓
2,talk,8,2,2,Skip↓
2,talk,9,3,3,Skip↓
2,talk,10,3,2,Skip↓
2,talk,11,3,1,Over↓
2,talk,12,4,1,Over↓
2,talk,13,4,3,Over↓
2,talk,14,4,2,Over↓
2,vote,1,2↓
2,vote,2,3↓
2,vote,3,2↓
2,execute,2,WEREWOLF↓ ... ..
```

リスト 6. パワープレイ阻止失敗例

```
2,talk,0,0,1,COMINGOUT Agent[01] WEREWOLF↓
2,talk,1,0,3,VOTE Agent[04]↓
2,talk,2,0,4,VOTE Agent[01]↓
2,talk,3,1,3,VOTE Agent[01]↓
2,talk,4,1,1,Skip↓
2,talk,5,1,4,Skip↓
2,talk,6,2,4,Skip↓
2,talk,7,2,1,Skip↓
2,talk,8,2,3,COMINGOUT Agent[03] WEREWOLF↓
2,talk,9,3,4,Over↓
2,talk,10,3,3,Skip↓
2,talk,11,3,1,Over↓
2,talk,12,4,4,Over↓
2,talk,13,4,3,Skip↓
2,talk,14,4,1,Over↓
2,talk,15,5,1,Over↓
2,talk,16,5,3,Over↓
2,talk,17,5,4,Over↓
2,vote,1,3↓
2,vote,3,1↓
2,vote,4,3↓
2,execute,3,VILLAGER↓
```

リスト 5, リスト 6 はともに人狼と村人のエージェント両方が自らを人狼であるとカミングアウトしており, リスト 5 では, 裏切り者が人狼に投票することによって人狼を追放することに成功し, リスト 6 では逆に裏切り者が村人に投票することで, 村人が追放されている。

また, カミングアウトには特殊な事例があり, リスト 7 に示す。リスト 7 では, 村人エージェントのみが自らは人狼であるとカミングアウトし, そのまま裏切り者, 人狼の票を集めて追放されている。裏切り者は本来カミングアウトした人狼への投票は避けるべきだが, このエージェントはそこまで至っていないと考えられる。

リスト 7. パワープレイ阻止失敗例(特殊)

```
2,status,1,WEREWOLF,ALIVE,AAA↓
2,status,2,POSSESSED,ALIVE,CCC↓
2,status,3,SEER,DEAD,BBB↓
2,status,4,VILLAGER,ALIVE,EEE↓
2,status,5,VILLAGER,DEAD,DDD↓
2,talk,0,0,4,VOTE Agent[01]↓
2,talk,1,0,1,VOTE Agent[04]↓
2,talk,2,0,2,Skip↓
2,talk,3,1,4,COMINGOUT Agent[04] WEREWOLF↓
2,talk,4,1,2,Skip↓
2,talk,5,1,1,Skip↓
2,talk,6,2,1,Skip↓
2,talk,7,2,4,Skip↓
2,talk,8,2,2,Over↓
2,talk,9,3,4,Skip↓
2,talk,10,3,1,Over↓
2,talk,11,3,2,Over↓
2,talk,12,4,1,Over↓
2,talk,13,4,2,Over↓
2,talk,14,4,4,Over↓
2,vote,1,4↓
2,vote,2,4↓
2,vote,4,1↓
2,execute,4,VILLAGER↓
```

3.5.2 投票先の変化

投票先の変化を以下のリスト 8, リスト 9 に示す。また, リスト 8 のゲームをゲーム 3, リスト 9 のゲームをゲーム 4 とし, それぞれの役職配置を表 10, 表 11 に示す。

表 10. ゲーム 3 役職配置

エージェント 2	WEREWOLF
エージェント 3	VILLAGER
エージェント 5	POSSESSED

表 11. ゲーム 4 役職配置

エージェント 3	VILLAGER
エージェント 4	WEREWOLF
エージェント 5	POSSESSED

リスト 8. 変更前投票例

```

2,vote,2,3↓
2,vote,3,5↓
2,vote,5,3↓
    
```

リスト 9. 変更後投票例

```

2,vote,3,4↓
2,vote,4,5↓
2,vote,5,4↓
    
```

リスト 8 の場合は人狼が村人に投票し、村人は人狼に投票、裏切り者は村人に投票し、結果として最も得票数の多いのは村人となっている。しかしリスト 9 では裏切り者であるエージェント 5 が人狼であるエージェント 4 に投票することで、人狼の得票数を最も高くすることに成功している。

これは文字通り裏切り者を惑わせることに成功したこととなり、パワープレイの阻止が成功していることを意味する。なお、ここで上げたのはあくまで一例であり、リスト 7 のようにパワープレイの阻止に失敗したパターンも存在する。

4 考察

4.1 勝率について

ここでは勝率の変化が予想と異なる方向に働いた点について考察する。勝率の変化に最も影響があるのは村人によるカミングアウトの変化だと考えたので、その点について確認することとする。村人のカミングアウトの変化についてはリスト 5, リスト 6, リスト 7 のとおりであるが、実際のところリスト 7 のように村人が単独でカミングアウトをしてしまうケースがあまりにも多く、かえって村人陣営の勝率が低下しているのではないかと考えられる。

以下の表 12 に 10000 回ゲームを実行し、そのうちで最終日の条件を満たしつつ村人が単独でカミングアウトした回数を示す。

表 12. 村人の単独カミングアウト回数

ゲーム数	カミングアウト回数
122	76

村人エージェントのみが人狼とカミングアウトしている原因について、それはカミングアウトのタイミングを実行学習にのみ委ねたからだと思われる。もしも人狼陣営がパワープレイを仕掛けた後にのみ村人陣営が阻止を図った場合、パターンの識別から後にカミングアウトした者が必ず追放されると考えて実行学習にのみタイミングを委ねたが、結果としては村人エージェントが単独で人狼とカミングアウトし、自陣の勝率を下げる結果となった。

またリスト 7 のゲームを X 戦目とした場合、直近の村人陣営のカミングアウトの有無と勝敗は以下の表 13 に示すとおりである。

表 13. 直近の村人陣営の動き

ゲーム	カミングアウト	村人陣営の勝敗
X-20 戦目	あり	勝利
X-19 戦目	なし	勝利
X-18 戦目		
X-17 戦目		
X-16 戦目		
X-15 戦目	なし	敗北
X-14 戦目		
X-13 戦目	なし	敗北
X-12 戦目	あり	敗北
X-11 戦目	あり	敗北
X-10 戦目	なし	勝利
X-9 戦目		
X-8 戦目	なし	敗北
X-7 戦目	あり	勝利
X-6 戦目	あり	敗北
X-5 戦目		
X-4 戦目	なし	敗北
X-3 戦目		
X-2 戦目	あり	勝利
X-1 戦目	なし	敗北

村人が生き残っていないか、初日で終了したゲームには斜線を引いた。

表 13 によれば、村人が発言できる状況が 13 回に対して、村人が人狼だとカミングアウトした場合の勝利パターンが 3、カミングアウトせずに敗北したパターンが 5 と、村人が発言できる状況下において、人狼だとカミングアウトした場合の勝率向上の検証を行うのは自然だと考えられる。

4.2 事前学習について

3.1 で述べた通り、事前学習をさせようとして大会に提出されていると仮定する。そのうえで村人陣営がパワープレイの阻止をする命令が組み込まれなかった理由について、条件を満たす事例があまりに少ないからだと考えられる。表 5, 表 6 にある通り、30000 ゲームに対してパワープレイに至るパターンが変更前では 1.9%、変更後は 1.4% と少数であったため、事前学習がおろそかになり、村人がカミングアウトするという結果に至らなかった可能性が高いと思われる。なお、人間規模の人狼ゲーム大会である「人狼 HOUSE」では「パワープレイの阻止」を行う事例がいくつもあり、人間は人狼ゲームのルールから演繹して戦術を組み立てることが可能であると推測され、そこに人工知能と

人間の違いが存在すると考えられる。

5 結言

本研究の当初の目的は、人工知能が学習困難な事項を明らかにすることであった。よって、人間が行うが人狼知能が行わない戦術を人狼知能に実装し、挙動や勝率などの変化を観察した。結果、当初のパワープレイを村人が阻止するという動きを達成することはできていたものの、勝率は予想と逆に変化した。しかし、3.4.2のように状況を限定することでパワープレイの阻止が村人陣営の勝率向上に寄与できていることが分かった。また、人間はルールから演繹して戦術を立てることができるのに対して、人工知能は今回のようなレアケースに対しての事前学習がおろそかになりがちである。したがって今回の戦術が組み込まれていなかったと考えられる。

今回検証した戦術は人狼を見つけ出す、というゲームの目的とは少々脱線しているものの、勝利に向かって食欲に動き、時には一般に言われている定石の裏を突くというものである。今現在は「人間らしく動く」という既存の人工知能には無い要素が人狼ゲームにかかわらず求められている。その中で私が解明できた点は必ずしも多くはないが、人工知能の発展に多少なりとも寄与できたのではないかと思われる。

謝辞

本研究に際して、様々なご指導を頂きました指導教員の西村俊二先生に深謝いたします。また、この研究の機会をくださった情報工学科の先生方、ゲームログを使用させていただきました cnd1 様、そして多くの知識やご指摘を下さいました同研究室の同期の皆様にも厚く御礼申し上げます。

参考文献

- [1] 鳥海不二夫・狩野芳伸・大槻恭士・園田亜斗夢・中田洋平・箕輪峻：人狼知能で学ぶ AI プログラミング，株式会社マイナビ出版，2017
- [2] 鳥海不二夫・片上大輔・大澤博隆・稲葉通将・篠田孝祐・狩野芳伸：人狼知能 一だます・見破る・説得する人工知能一，森北出版株式会社，2016
- [3] 人狼知能プロジェクト，人狼知能大会，http://aiwolf.org/aiwolf_contest，[アクセス日：1 8 2020].
- [4] 人狼知能プロジェクト，人狼知能プラットフォーム，<http://aiwolf.org/server>，[アクセス日：11 10 2019].