

平成 28 年度 卒業論文

人狼ゲームをプレイするエージェントの実装

奈良女子大学 理学部 情報科学科 4 回生 新出研究室  
学籍番号 : 13251391

池内 加奈

平成 29 年 2 月 13 日

## 概要

近年、人工知能の研究課題として人狼ゲームをプレイする人工知能、「人狼知能」が話題になっている。先行研究では強化学習を用いた研究が多く見受けられるが、本研究では論理的思考に着目し、BDI 論理に基づく論理的な人狼知能の実現を目指した。また、最後にはより論理的な人狼知能を実現するための課題について述べた。

## 目次

<b>1</b>	<b>はじめに</b>	<b>2</b>
1.1	研究背景	2
1.2	人狼ゲームのルール	2
<b>2</b>	<b>人狼知能プラットフォーム</b>	<b>4</b>
2.1	人狼知能サーバ	4
2.2	人狼知能プロトコル	5
<b>3</b>	<b>実装</b>	<b>6</b>
3.1	実装方針	6
3.2	Java と Jason の通信	6
3.3	実装	7
3.3.1	情報処理メソッド	8
3.3.2	対象指定メソッド	10
3.3.3	会話メソッド	11
3.3.4	命名メソッド	11
3.3.5	その他のメソッド	11
<b>4</b>	<b>戦略</b>	<b>12</b>
4.1	偽人狼の設定	12
4.2	投票先の選択	12
4.3	襲撃先の選択	13
4.4	偽占い(霊能)結果の設定	13
<b>5</b>	<b>結果</b>	<b>14</b>
5.1	実行結果	14
5.2	実験と結果	15
<b>6</b>	<b>おわりに</b>	<b>16</b>

# 1 はじめに

## 1.1 研究背景

近年、人工知能の研究課題として人狼ゲームをプレイする人工知能、「人狼知能」が話題になっている。人狼ゲームは対話型のコミュニケーションによって進行する不完全情報ゲームであり、その特徴から人工知能の標準問題としての可能性も検討されている [1]。また、人狼ゲームをプレイするエージェントには人工知能の観点から多岐に渡った技術が必要であるため、人狼知能の研究が進められることは人工知能の技術の発展にも貢献することができるものと考えられる。

先行研究では、強化学習を用いたものが多く存在するが、論理的思考に着目したものはあまり見受けられない。一方で、[2] において BDI 論理を用いたエージェントの推論記述法が提案されている。BDI 論理とは、通常の論理式に信念、願望、意図を表す演算子を加えて人間の心理状態を表せるようにしたものである。[2] では実際の人狼ゲームでのプレイヤーの推論を信念や意図といった概念を用いて記述しており、これらの概念を持つことが人狼ゲームをプレイするエージェントにとって有利に働くと考えられる。そこで、我々は BDI 論理をベースとする人狼知能エージェントの実現を考える。BDI 論理は BDI アーキテクチャーのもとになっている論理モデルであるため、BDI アーキテクチャーを基礎とした処理系 Jason[4] を用いることで、論理的思考に基づく人狼知能の実現を目指した。

人狼知能エージェント本体の実装については人狼知能プロジェクト [3] によって人狼知能プラットフォームが公開されているなど開発環境がある程度整っているため、それを用いる。

## 1.2 人狼ゲームのルール

人狼ゲームのルールには様々なものが存在するが、本論文では人狼知能プラットフォームに従って本研究で用いたルールについて述べる。

まず、ゲーム開始時に各プレイヤーに 6 つの役職が割り当てられる。ゲームは昼のフェーズと夜のフェーズを繰り返すことによって進行し、昼のフェーズでは全プレイヤーでの対話と人狼のみの対話、夜のフェーズでは投票と各プレイヤーの行動を行う。各プレイヤーの行動では役職ごとに与えられた特殊な能力をそれぞれ発揮する。表 1 に各役職の詳細とプレイヤーが 15 人の場合の各役職の人数を示す。

役職ごとに陣営が決められており、各プレイヤーは自陣営の勝利を目的としてゲームを進める。各陣営の勝利条件は以下のとおりである。

- 村人陣営 (村人、占い師、霊能者、狩人) — 人狼をすべて追放する
- 人狼陣営 (狂人、人狼) — 村人の数を人狼の数以下にする<sup>1</sup>

---

<sup>1</sup>このとき狂人は村人としてカウントされる。

ゲームが進むにつれ、各プレイヤーは投票で追放されるか、人狼に襲撃されて死亡することでゲームから除外されていく。そうして勝利条件のどちらかが達成されたとき、ゲームは終了する。

	役職	人数 (人/15 人)	能力
村人陣営	村人	8	なし
	占い師	1	1 日の終わりに 1 人のプレイヤーを占い、そのプレイヤーが人間かどうか知ることが出来る
	霊能者	1	プレイヤーが追放された場合にそのプレイヤーが人間かどうか知ることが出来る
	狩人	1	1 日の終わりに 1 人のプレイヤーを指定して人狼の襲撃から守ることが出来る
人狼陣営	狂人	1	なし 人狼陣営だが占い結果は人間
	人狼	3	1 日の終わりに 1 人のプレイヤーを指定して襲撃することが出来る

表 1: 与えられる役職

## 2 人狼知能プラットフォーム

人狼知能プラットフォームとは、人狼知能プロジェクト [3] が公開している人狼知能エージェントの開発環境を整えるためのサーバとプロトコルである。人狼知能プロジェクトホームページにて jar ファイルが配布されている他、GitHub でソースコードも公開されている。

### 2.1 人狼知能サーバ

人狼知能プラットフォームでは、1つのサーバにクライアント(プレイヤー)をTCP/IP通信で複数接続することによってゲームを実行する。サーバはクライアントに Request を通知し、その際必要に応じてゲームの状態や会話内容などの情報を提供する。クライアントはサーバから Request と情報を受け取り、必要に応じてサーバに返信する。ゲームはこの繰り返しで進行する(図1)[5]。

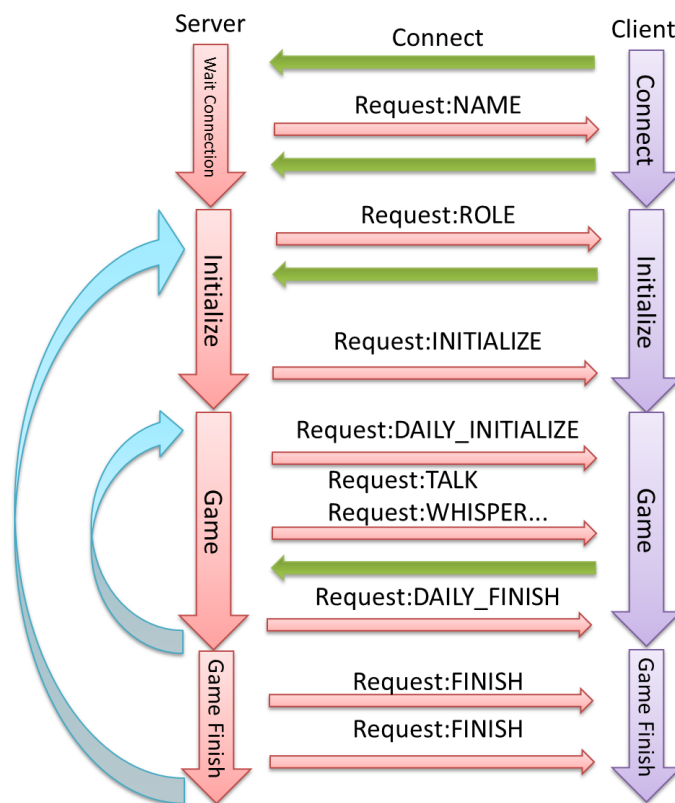


図 1: サーバとクライアントの通信 [5]

## 2.2 人狼知能プロトコル

人工知能に自然言語によるゲームを行わせるには課題が多い [6] ため、人狼知能プラットフォームでは人狼専用の対話プロトコルが用意されている。人狼知能プラットフォーム ver0.4.4 においてエージェントが行える発話は以下のとおりである [7]。

1. estimate : target の役職は role だと思う。  
例 : ESTIMATE Agent[01] SEER
2. comingout : target の役職は role だ。  
例 : COMINGOUT Agent[01] MEDIUM
3. divination : target を占う。  
例 : DIVINATION Agent[01]
4. divined : target を占った結果 result (人間 or 人狼) だった。  
例 : DIVINED Agent[01] HUMAN
5. identified : target は霊媒の結果 result (人間 or 人狼) だった。  
例 : IDENTIFIED Agent[01] WEREWOLF
6. guard : target を護衛する。  
例 : GUARD Agent[01]
7. guarded : target を護衛した。  
例 : GUARDED Agent[01]
8. vote : target に投票する。  
例 : VOTE Agent[01]
9. attack : target に襲撃投票する。  
例 : ATTACK Agent[01]
10. agree : talkDay 日目, 種類 talkType, talkID 番目の発話に同意する。  
例 : AGREE TALK day3 ID:20
11. disagree : talkDay 日目, 種類 talkType, talkID 番目の発話に反対する。  
例 : DISAGREE WHISPER day2 ID:10
12. request: agent に content を要求する。  
例 : REQUEST(Agent[01] DIVINE Agent[02]) (Agent[01] に Agent[02] を占うことを要求する)  
REQUEST(DIVINE Agent[02]) (Agent[02] を占うことを要求する)
13. over : もう話すことは無い。
14. skip : 様子を見る。

### 3 実装

#### 3.1 実装方針

本研究で扱う人狼ゲームには役職が6つ存在し、役職ごとに行動等の実装を行う必要があるが、本研究では人狼のみの実装を行った。実装は人狼知能プラットフォームのサンプルプログラムをもとに行い、思考部分を Jason へ移植した。また、サンプルプログラムの戦略に加えて新たな戦略の追加を行った。

#### 3.2 Java と Jason の通信

人狼知能プラットフォームをはじめとしたエージェントの実装は Java 言語を用いるため、Jason に思考部分を移植するにあたって Java と Jason の間で情報のやりとりをする必要がある。よって、Java と Jason の間で通信を行った。本研究では襲撃先の選択、投票先の選択、偽占い(霊能)結果の設定を Jason で思考する部分とした。よって、エージェント(Java プログラム)のメソッド vote、attack、getFakeJudge 内それぞれで Jason と通信して情報の送受信を行い、Jason でそれぞれの行動についての思考を行うようにした。

各メソッドと Jason のやり取りの流れは次の通りである。

1. エージェントが思考に必要な情報をリストとしてひとつにまとめて文字列に変換し、Jason に送信
2. Jason はエージェントから受信した情報をもとに思考し、結果を文字列に変換してエージェントに送信
3. エージェントが Jason から受信した情報を型変換して返り値として返す

これにより、エージェントの各メソッドでは情報の送受信のみを行い、Jason に思考を担わせることができる。

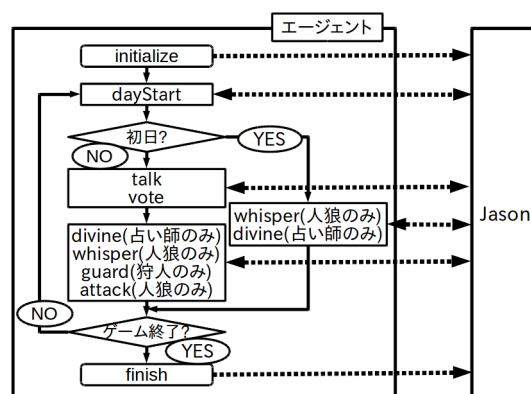


図 2: 実装の流れ

### 3.3 実装

人狼知能エージェントの実装は人狼知能プラットフォームの Player インターフェースを継承して行う。Player インターフェースを継承したクラスは以下の 11 個のメソッドを実装する必要がある [8]。

- 情報処理メソッド : initialize, update, dayStart, finish
- 対象指定メソッド : vote, attack, guard, divine
- 会話メソッド : talk, whisper
- 命名メソッド : getName

本章ではこれらのメソッドと getFakeJudge メソッド、そして Jason プログラムの実装について述べる。ただし、人狼のプレイヤーでは実装する必要のないメソッド guard, divine については省くこととする。

実装においては Jason と通信するためのソケットを生成しており、そのソケットを内部に持つオブジェクトが変数 ch に代入されていて、このオブジェクトのメソッド writeLine() や readLine() などと呼ぶことで Jason と通信が行われている。

### 3.3.1 情報処理メソッド

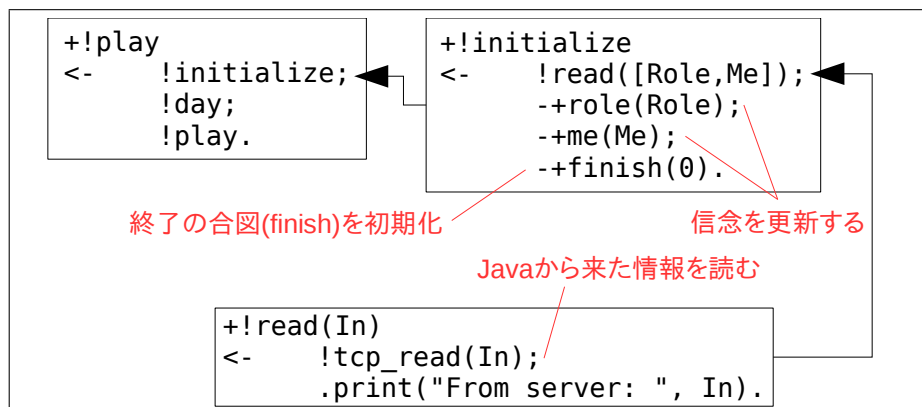
情報を処理するためのメソッドで、戻り値はない。

- **initialize(GameInfo,GameSetting)**

ゲーム開始時に1度だけ呼ばれるメソッドで、ゲームの状態 (GameInfo) と設定 (GameSetting) が与えられる。騙る役職やカミングアウトのタイミングを決定する (人狼のみ)。Jason と通信するためのソケットを生成し、Jason へ自分の役職とエージェント番号を送信する。

```
//ソケットの生成
ch = new tcpSrv(port);
...
//情報の送信
String str = "["+werewolf"+","+myidx+"]";
ch.writeLine(str);
```

Jason では情報を受け取って信念を更新する。



- **update(GameInfo)**

initialize 以外の全てのメソッドの前に呼ばれる。ゲームの情報を引数 (GameInfo) や会話から得て情報を更新する。狂人がどのエージェントかを予想し、更新する (人狼のみ)。

- **dayStart()**

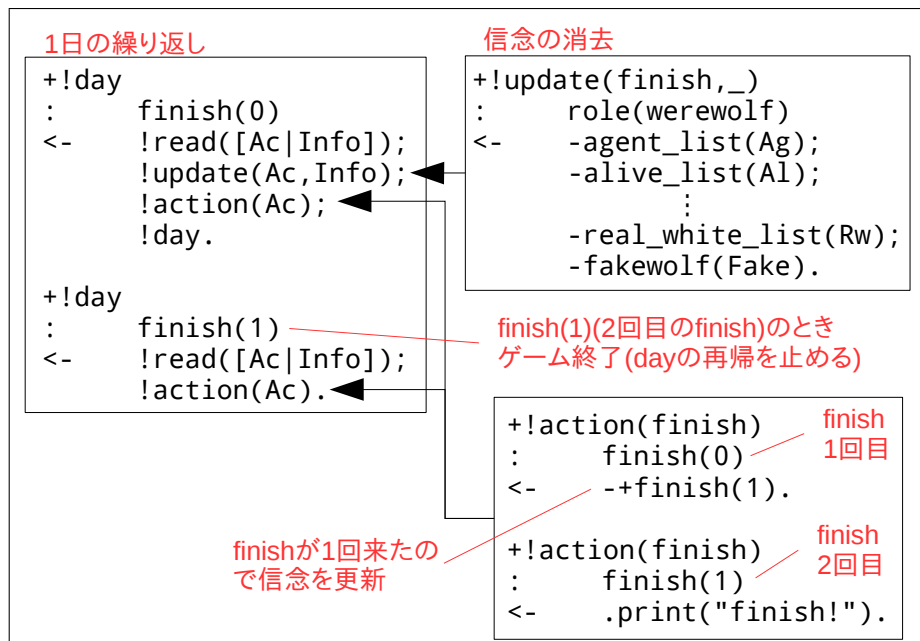
1日の始めに1度だけ呼ばれる。使用する変数の初期化等を行う。

- finish()

ゲーム終了時に2度呼ばれる。Jasonへ終了の合図を送る。

```
//終了の合図として finish という文字列を送信
String str = "["+finish+"]";
ch.writeLine(str);
```

Jasonでは、新しいゲームで古い信念が使用されないように信念の消去を行う。



### 3.3.2 対象指定メソッド

戻り値として対象となる Agent を返す。

- vote()

その日に投票する対象プレイヤーを返す。投票先を選択して変数に代入するメソッド chooseVoteCandidate() で設定された投票先、voteCandidate を用いて投票先を最終的に決定する。

```
void chooseVoteCandidate(){
    //リストをインデックスのリストに
    List<Integer> agentlist = new ArrayList<Integer>();
    for(Agent item : currentGameInfo.getAgentList()){
        agentlist.add(item.getAgentIdx());
    }
    ...
    //Jason へ文字列を送信
    str = "["+"vote"+" "+agentlist+" "+...+" "+fakeroles+"";
    ch.writeLine(str);
    //Jason から文字列を受け取る
    String inStr = ch.readLine();
    ...
    //受け取った文字列を変換し、変数に代入
    int i = Integer.parseInt(inStr);
    voteCandidate = Agent.getAgent(i);
}
```

Jason 側

```
+!action(vote)
: info(Ag,Al,Co,...,werewolf)
<- ?villager(Al,Vil); //生存者から人狼陣営を削除
    ?fakewolf(Fake); //偽人狼
    .intersection(Al,Fake,FA); //生存している偽人狼
    ...
}else{
    ?sort_doubtful(Candidate,Out);
    Out = [Target|_];
}
!send(Target). ←
```

受け取った情報

候補者リスト (Candidate)から Targetを1人選ぶ

```
+!send(Out)
<- .print("To server: ",Out);
    tcp_write(Out).
```

Javaに結果を送る

後述する attack,getFakeJudge でも大枠は同じである。

- **attack()** (人狼のみ)  
その日に襲撃投票するプレイヤーを返す。襲撃先を選択して変数に代入するメソッド `chooseAttackCandidate()` で設定された襲撃先、`attackCandidate` を用いて投票先を最終的に決定する。

### 3.3.3 会話メソッド

戻り値として発言する内容を `String` 型で返す。

- **talk()**  
村全体に対して発言する内容を返す。投票先を選択して変数に代入するメソッド `chooseVoteCandidate()` を内部で呼び出し、発言に反映する。
- **whisper()** (人狼のみ)  
人狼だけに対して発言する内容を返す。襲撃先を選択して変数に代入するメソッド `chooseAttackCandidate()` を内部で呼び出し、発言に反映する。

### 3.3.4 命名メソッド

- **getName()**  
プレイヤーの名前を `String` 型で返す。

### 3.3.5 その他のメソッド

- **getFakeJudge()**  
引数として自分が騙る役職を与え、偽占い(霊能)結果を `Judge` 型で返す。各所必要なメソッドで呼ばれる。

## 4 戦略

本章では人狼知能がゲームを進める上での戦略について、Jason で実装したものを述べる。なお、4.1 の戦略は新たに追加したもので、それ以外の戦略は人狼知能プラットフォームのサンプルプログラムを参考に 4.1 の概念を加えたものである。

### 4.1 偽人狼の設定

人狼のプレイヤーは自分が人間だと偽る必要があるため、偽の人狼を仕立て上げる必要がある。各プレイヤーに 0.0~1.0 の偽の疑わしさを設定し、いちばん初めに村人陣営と人狼陣営が同じ役職で CO(カミングアウト)<sup>2</sup> したとき偽の疑わしさをもとに偽の人狼を設定する。偽の疑わしさは Jason のルールとして論理式で記述した。

```
//エージェント A の (偽の) 疑わしさ D
//人狼陣営
agent(A)[doubtful(D)] :- role(werewolf)
                        & werewolf_list(We) & .member(A,We) & D=0.0.
agent(A)[doubtful(D)] :- role(werewolf)
                        & possessed(Po) & Po \== -1 & A=Po & D=0.0.
...
```

### 4.2 投票先の選択

生存者のうち、

- 村人騙りのとき
  1. 偽人狼
  2. 村人陣営全体
- 占い師、霊能者騙りのとき
  1. 偽人狼、対抗 CO<sup>3</sup> 者
  2. 自分が白判定を出していないプレイヤー
  3. 村人陣営全体

の優先順位で候補を選び、そこから最も偽の疑わしさが高いプレイヤーを選ぶ。ただし、以前選択したプレイヤーが候補に入っている場合はそのプレイヤーを選ぶ。

<sup>2</sup>自分が能力者であることを公表すること。必ずしも真実とは限らない。

<sup>3</sup>対抗カミングアウト。名乗り出た能力者に対して、自分が本物だと対抗すること。

### 4.3 襲撃先の選択

生存者のうち、

1. 偽人狼以外の真占い師
2. 偽人狼以外の真霊能者
3. 自分も真占い師も白判定<sup>4</sup>を出したプレイヤー
4. 1~3 と偽人狼以外の村人陣営
5. 村人陣営全体

の優先順位で候補を選び、そこから最も偽の疑わしさが高いプレイヤーを選ぶ。ただし、以前選択したプレイヤーが候補に入っている場合はそのプレイヤーを選ぶ。1~4において偽人狼が選択肢から除外されているのは、人狼同士で襲撃することができないために偽の人狼を襲撃すると言動が破綻する恐れがあるからである。

### 4.4 偽占い(霊能)結果の設定

偽人狼に黒判定<sup>5</sup>を出し、偽人狼が決定するまでは白判定を出しつづける。

---

<sup>4</sup>人間判定。対義語は「黒判定」

<sup>5</sup>人狼判定。対義語は「白判定」

## 5 結果

### 5.1 実行結果

ゲームを実行した結果、Jason 側の画面では Jason が情報を受け取って思考結果を Java に送信していることが確認出来た。[図 3]

自分の役職、エージェント番号を受信

情報(行動の種類、全エージェントの番号のリストなど)を受信

```
[client] From server: [werewolf,15]
[client] From server: [fakejudge,[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15],[1
[client] To server: -1,HUMAN
[client] From server: [fakejudge,[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15],[1
[client] To server: -1,HUMAN
[client] From server: [fakejudge,[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15],[1
[client] To server: -1,HUMAN
```

図 3: 実行結果 1

また、ゲーム実行時に作成される log ファイルでゲームの進行過程および最終結果も確認することが出来た。[図 5]

```
1,talk,4,0,5,VOTE Agent[09]
1,talk,5,0,8,VOTE Agent[02]
1,talk,6,0,10,VOTE Agent[13]
1,talk,7,0,13,COMINGOUT Agent[15]
1,talk,8,0,15,VOTE Agent[01]
1,talk,9,0,9,VOTE Agent[02]
1,talk,10,0,2,VOTE Agent[10]
1,talk,11,0,6,VOTE Agent[04]
1,talk,12,0,1,VOTE Agent[06]
1,talk,13,0,4,VOTE Agent[09]
1,talk,14,0,7,VOTE Agent[08]
1,talk,15,1,15,REQUEST(VOTE Agent[01])
1,talk,16,1,6,REQUEST(VOTE Agent[02])
1,talk,17,1,9,REQUEST(VOTE Agent[03])
1,talk,18,1,7,Skip
```

1日目の8番目の、15番のエージェント  
(Jasonを使用したエージェント)の発言:  
1番のエージェント(Agent[01])に投票する

図 4: 実行結果 2 (ゲームの進行過程)

```

7,status,1,VILLAGER,DEAD,Sample09
7,status,2,VILLAGER,DEAD,Sample08
7,status,3,VILLAGER,DEAD,Sample07
7,status,4,VILLAGER,DEAD,Sample06
7,status,5,VILLAGER,ALIVE,Sample12
7,status,6,WEREWOLF,ALIVE,Sample11
7,status,7,MEDIUM,DEAD,Sample10
7,status,8,BODYGUARD,DEAD,Sample05
7,status,9,WEREWOLF,DEAD,Sample04
7,status,10,SEER,DEAD,Sample15
7,status,11,VILLAGER,DEAD,Sample03
7,status,12,VILLAGER,DEAD,Sample14
7,status,13,POSSESSED,DEAD,Sample02
7,status,14,VILLAGER,DEAD,Sample13
7,status,15,WEREWOLF,ALIVE, Jason01
7,result,1,2,WEREWOLF

```

7日目(ゲーム終了時)の  
15番のエージェントの状態:  
人狼、生存、名前はJason01

ゲームの結果:  
生存者が人間1人、人狼2人で  
人狼の勝ち

図 5: 実行結果 2 (ゲームの結果)

## 5.2 実験と結果

人狼のうち1人を Jason を使用したエージェントとした場合と、全てをサンプルプログラムそのままのエージェントとした場合で人狼陣営の勝率を比較した。Jason を使用したエージェントは人狼しか実装していないため役職は人狼で固定し、5人、15人でゲームをそれぞれ100回ずつ実行した。その結果、勝率の差はほとんどみられなかった。よって現時点で Jason を使用したエージェントはサンプルエージェントとほぼ同等の強さであると言える。強さに差が出なかった要因の1つとして、戦略の甘さが考えられる。例えば、自分以外の人狼の偽占い結果が行動に加味されないため、自分が村人騙りをした場合の選択肢が少なくなることなどが挙げられる。

人狼の内訳	人狼陣営の勝率 (%)
Jason×1	48
サンプル×1	50

表 2: 5人のとき (うち人狼1人)

人狼の内訳	人狼陣営の勝率 (%)
Jason×1, サンプル×2	78
サンプル×3	74

表 3: 15人のとき (うち人狼3人)

## 6 おわりに

本研究では、BDI 論理を用いた論理的思考に基づく人狼知能の実現を目指した。エージェントの思考部分を Jason に担わせることで、論理的思考に基づいた行動をさせることができた。

今後の課題として、まず人狼以外の役職の実装が挙げられる。本研究では人狼のみの実装を行ったため、他の役職にも対応していく必要がある。次に、戦略の改善が挙げられる。思考部分の戦略に甘さが存在することが問題となっているため、より強い戦略に改善する必要がある。また、サーバとの応答時間の上限を緩めているため、Jason での思考時間や Jason の実行開始にかかる時間を短縮する必要がある。最後に、発言内容への思考の反映が挙げられる。現在、Jason での思考は 3 つの限られたメソッドでしか行われていないため、発言内容に関しても Jason で思考することが必要である。これらの課題を解決することで、より強い、論理的思考に基づく人狼知能に近づけることが望まれる。

## 謝辞

本研究を進めるにあたり、丁寧かつ熱心なご指導を頂いた新出尚之准教授に心より感謝申し上げます。ありがとうございました。

## 参考文献

- [1] 篠田孝祐, 鳥海不二夫, 稲葉通将, 大澤博隆, 片山大輔. 人工知能標準問題としての人狼ゲームの提案. 第24回インテリジェント・システム・シンポジウム, pp. 74-77, 2014.
- [2] 大澤博隆, 鳥海不二夫, 稲葉通将, 片山大輔, 梶原健吾, 篠田孝祐. 人狼知能達成におけるエージェントの推論モデル. In *The 19th Game Programming Workshop 2014*, pp. 157-161, 2014.
- [3] 鳥海不二夫, 稲葉通将, 大澤博隆, 片山大輔, 篠田孝祐, 松原仁, 狩野芳伸, 大槻恭士, 園田亜斗夢. 人狼知能プロジェクト. <http://aiwolf.org/>. 人狼知能プロジェクトホームページ.
- [4] Rafael H. Bordini, Jomi Fred Hübner, and Michael Wooldridge. *Programming Multi-Agent Systems in AgentSpeak using Jason*. John Wiley & Sons, 2007.
- [5] 人狼サーバクライアントプロトコル説明 ver0.3.1. <http://www.aiwolf.org/aiwp/wp-content/uploads/2014/03/サーバクライアントプロトコルver0.3.1.pdf>.
- [6] 鳥海不二夫, 梶原健吾, 大澤博隆, 稲葉通将, 片山大輔, 篠田孝祐. 人狼知能サーバの構築. In *The 19th Game Programming Workshop 2014*, pp. 127-132, 2014.
- [7] 0.4.4 での発話生成の方法. <http://aiwolf.org/control-panel/wp-content/uploads/2014/03/0.4.4での発話生成の方法.pdf>.
- [8] 人狼知能ブレ大会@gat2017 レギュレーション. <http://aiwolf.org/control-panel/wp-content/uploads/2017/01/人狼知能ブレ大会@GAT2017レギュレーションVer1.00-20170106b.pdf>.