

理由に基づく協調要請を行える人狼エージェント

奈良女子大学 生活環境学部 情報衣環境学科
生活情報通信科学コース 4回生
新出研究室 19480346 中谷美友

2023年2月5日

概要

近年、人工知能の研究として人狼ゲームをプレイする人狼知能が注目されている。先行研究では強化学習を用いたものが多く、論理的思考で行動を決めているものはあまり見受けられない。我々は論理的思考が可能であり、それによって人間らしい行動をとる人狼エージェントの実現を目指している。本研究では人狼知能プロトコルに規定されている Because 文を利用する機能をエージェントに導入し、これを用いて要請を行ったり、要請を吟味して応じたりすることで、協力プレイを行えるエージェントを作成した。本論文ではこれについて述べ、さらに、より人間らしい行動をとる人狼エージェントを実現するための課題を述べる。

目次

1	はじめに	2
1.1	研究背景	2
1.2	人狼ゲームのルール	2
2	人狼知能プラットフォーム	4
2.1	人狼知能サーバ	4
2.2	人狼知能プロトコル	4
3	実装	5
3.1	Because 文の導入	5
3.2	Jason エージェントの複数利用	6
4	結果	7
4.1	実行結果	7
4.2	実験結果	9
5	まとめ	9

1 はじめに

1.1 研究背景

近年、人工知能の研究として、人狼ゲームをプレイする人狼知能が注目されている。人狼ゲームは対話型のコミュニケーションゲームによって進行する不完全情報ゲームである。この特徴から、人工知能の標準問題としての可能性が検討されている [1]。人狼ゲームを行うエージェントには、人工知能の観点から推論やコミュニケーションなど様々な技術が必要である。そのため、人狼知能の研究が進められることによって、人工知能の技術の発展に貢献することができると考えられている。

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。そのなかで、大澤ら [2] は、人狼におけるエージェントの推論モデルとして、人間の心的状態を表せる BDI 論理を用いることによって、実際の人狼ゲームで行われている複雑な推論が BDI 論理を用いて記述できることを示している。また、論理的思考を行えるエージェントが十分な強さで実現すれば、その思考や行為決定の過程を後で出力することで、ゲーム後のエピソードを楽しむことができると期待できる [3]。我々は、BDI 論理に基づいた論理的思考が可能であり、かつ人間らしい行動をとる人狼エージェントの実現を目指している。

当研究室の先行研究 [4] では、エージェント本体の実装に人狼知能プロジェクト [5] によって公開されている人狼知能プラットフォームを用い、エージェントの思考部分に BDI エージェントの実装プラットフォームである Jason [6] を用いることによって、論理的思考に基づいた行動が可能となっている。人間同士が人狼ゲームを行う際には、発言から怪しい人を見つけることや、周りを説得し、ある特定の人に投票を集めるなどの協力プレイを行うことができる。このような人間らしいエージェントを作成するための過程として、本研究では人狼知能プロトコルに規定されている Because 文を用いて理由を発言し、他のエージェントの Because 文の発言により行動を変えることで協力プレイを可能にすることを目指した。

1.2 人狼ゲームのルール

人狼のゲームのルールには様々なものがあるが、本論文では、本研究で用いた人狼知能プラットフォームの 15 人ルールについて説明する。

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、村人陣営と人狼陣営に分けられる。各プレイヤーは自陣営の勝利を目標としてゲームを進める。村人陣営の勝利条件は人狼を全滅させること、人狼陣営の勝利条件は村人の数を人狼の数以下にすることである。

ゲームは昼と夜の 2 つのフェーズを繰り返すことによって進行していく。昼のフェーズは生存プレイヤーが対話を行い、投票によって 1 人が追放される。夜のフェーズは、占い師や霊能者、狩人が各役職に与えられた能力を行使する。そして、人狼同士の対話によって決められた 1 人が襲撃される。村人陣営は、占い結果や霊媒結果などの情報や他のプレイヤーの発言から人狼を見つけることが重要である。一方人狼陣営側は村人陣営になりすまし、偽情報を発言しながら信用を得ることが重要である。

ゲームが進むにつれて、各プレイヤーは投票により追放されるか人狼に襲撃されることでゲームから除外されていき、村人もしくは人狼陣営どちらかが勝利条件を達成した時、ゲームは終了する。

表1 各役職の能力と人数

	役職	人数	判定	能力
村人陣営	村人	8人	人間	なし
	占い師	1人		1日の終わりに生存プレイヤーから1人を選んで、そのプレイヤーが人狼かどうか知ることができる
	霊能者	1人		1日の終わりに直前に追放されたプレイヤーが人狼であったかを知ることができる
	狩人	1人		1日の終わりに生存プレイヤーから1人選んで人狼の襲撃から守ることができる
人狼陣営	狂人	1人		なし ※人狼陣営だが判定結果は人間
	人狼	3人	人狼	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できる

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は、人狼知能プロジェクト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やプロトコルである。人狼知能プロジェクトのホームページで jar ファイルが配布されており、GitHub でソースコードも公開されている。Java、Python、C#が利用可能であるが、本研究では Java を用いて研究を行った。

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情報や会話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答する。ゲームはこの繰り返しで進行していく [7]。

2.2 人狼知能プロトコル

人狼知能において、自然言語を用いてゲームを行うのは難しいため、人狼知能プラットフォームは人狼の会話に必要なプロトコルが用意されている。1つの文章は役職、動詞、エージェント名の組み合わせから構成される。人狼知能プロトコル ver3.6[8] では because 文を含む 7つの演算子が追加された。以下に発話の例を挙げる。

- ESTIMATE
Agent[01] ESTIMATE Agent[02] WEREWOLF
-Agent[01] が Agent[02] を人狼と推測する。
- COMINGOUT
Agent[01] COMINGOUT Agent[02] WEREWOLF
-Agent[01] が Agent[02] を人狼と宣言する。
- VOTE
Agent[01] VOTE Agent[02]
-Agent[01] が Agent[02] に投票する。
- ATTACK
Agent[01] ATTACK Agent[02]
-Agent[01] が Agent[02] を襲撃する。
- DIVINED
Agent[01] DIVINED Agent[02]
-Agent[01] が Agent[02] を占う。
- BECAUSE
Agent[02] BECAUSE (DAY1 Agent[01] vote Agent[02])(vote Agent[01])
-1 日目に Agent[01] が Agent[02] に投票した。だから、(Agent[02] は) Agent[01] に投票する。

3 実装

3.1 Because 文の導入

本研究では、村人陣営側である村人、占い師、霊媒師、狩人のみを対象に、Because 文の導入を行った。各エージェントは下記の場合に Because 文を用いて推測発言や投票要請を行う。

- 占い師、霊媒師

- (1) 占い結果が人狼であった場合

”自分は占い師である”というカミングアウトと占い結果を理由とし、占い結果が人狼であったエージェントへの投票要請を行う

(例) 「2 日目に占ったら Agent[02] は人狼であった。だから、Agent[02] に投票する。かつ、誰かが Agent[02] に投票することを要請する。」

- (2) 自分の霊媒結果と異なる発言をした偽占い師がいた場合

自分の霊媒結果と偽占い師の占い結果との矛盾を理由とし、”偽占い師は人狼または狂人であると思う”という推測発言を行う

(例) 「Agent[01] の霊媒結果は人間だった。かつ、DAY1 に Agent[02] は Agent[01] を人狼だと占った。だから、Agent[02] は人狼もしくは狂人だと思う。」

- (3) 対抗 CO(自分の他に占い師または霊媒師だと名乗る人) が現れた場合

”自分は占い師(霊媒師)である”というカミングアウトと他のエージェントが同じ役職を名乗ったことを理由とし、”対抗 CO エージェントは人狼または狂人であると思う”という推測発言を行う

(例) 「私は占い師です。かつ Agent[01] は占い師と名乗った。だから、Agent[01] は人狼もしくは狂人だと思う。」

- 村人、狩人

- (1) 占い師と名乗るものから、人狼だと占われた場合

”自分は村人である”というカミングアウトと占い師と名乗るエージェントから人狼と占われたことを理由とし、”間違った占い結果を発言する偽占い師は人狼または狂人であると思う”という推測発言を行う

(例) 「私は村人です。かつ Agent[01] は、私を人狼だと占った。だから、Agent[01] は人狼もしくは狂人だと思う。」

また、本研究では他のエージェントが Because 文を用いて発言を行った場合、発言者と発言の対象者の怪しさの度合いによって行動を変化させた。エージェントの怪しさは先行研究 [9] で用いられた、投票度を基に測る。投票度は、役職推定をもとにした投票先を決定する値であり、高ければ高いほどそのエージェントを怪しんでいることを表す。本研究ではまず、各エージェントの発言に現れた投票先によって投票度を変化させる仕組みを導入した。具体的には、自分と同じ投票先のエージェントの投票度を減少させ、自分に投票したエージェントの投票度を増加させるようにした。その上で、発言者の投票度が発言の対象者の投票度より小さい場

合は、投票先をその発言の対象者に変更することで、行動を変化させた。複数のエージェントが Because 文を発言した場合には、最も投票度の低いエージェントの要請に従うこととする。

3.2 Jason エージェントの複数利用

先行研究では、Jason を用いたエージェントは、人狼知能サーバに接続するエージェント本体のプログラムと Jason による思考部分の TCP による通信において、ポート番号を 60000 に固定しており、その結果、通信路が 1 つのみであったため、複数のエージェントを使用することができなかった。そこで各エージェントが異なるポート番号を使えるように変更し、自分用の通信路を確保することで、複数のエージェントを使用することを可能にした。このことによって同じ戦略を持つエージェントを複数使用できるようになり、協力プレイを行うことが可能になる。図 1, 2, 3 にプログラムの変更点の主要箇所を示す。エージェント本体の側では、個別のプレイヤーの具象クラスで図 1 のように、自分が使うポート番号を指定し、親クラスが図 2 のように、指定したポートにつなぐ共通の処理を実装している。また、Jason 側でも図 3 のように、TCP 接続するアクションにおいて、接続するポート番号を指定できるようにしている。

```
package org.jasonaiwolf.player;
import org.jasonaiwolf.tcp.tcpSrv;

public final class JasonSeerA extends JasonSeer{
    // JasonSeerのfinalは撤去してJasonSeerをAbstractにする
    // AbstractRoleAssingPlayerでtcpSrvを呼んで通信路を作るのはやめた

    @Override
    protected int getport(){
        return 60000;
    }
}
```

図 1 JasonSeerA

```
// 各エージェントが自分用通信路を確保
protected abstract int getport();
// 各エージェントが自分が使うポート番号を返すメソッド
protected void connect(){
    port = getport();
    // System.err.println("Connecting to "+port);
    try {
        ch = new tcpSrv(port);
    } catch(IOException e) {
        System.err.println("IO Exception");
        System.exit(1);
    }
    // System.err.println("Connected to "+port);
}
```

図 2 SampleBasePlayer

```
@Override
public boolean executeAction(String ag, Structure action) {
    boolean result;
    //System.out.println("hahaha1");
    if (action.getFuncion().equals("tcp_connect")
        && action.getArity() == 1
        && action.getTerm(0).isNumeric()){
        //System.out.println("hahaha2");
        try{
            //System.out.println(ag+" hahaha3");
            int port = (int) (((NumberTerm) (action.getTerm(0))).solve());
            TcpComm ch = chMap.get(ag);
            if(ch != null){ // 接続済み
                // この場合、指定したポート番号が前回と違っていても
                // 再接続はしない
                //System.out.println("hahaha4");
            } else {
                //System.out.println(ag+" hahaha5 "+port);
                ch = new TcpComm(hostname, port);
                //System.out.println(ag+" hahaha6 "+port);
                chMap.put(ag, ch);
            }
        } catch(Exception e){
            logger.info("Connection failed");
            return false;
        }
    }
    result = true;
}
```

図 3 TcpEnv

4 結果

4.1 実行結果

図4は、ゲームを実行した結果のログである。あるゲームの2日目のログの中から発話制御に関する発言であるSKIPとOVERを除いて示した。エージェント番号1, 2, 4, 6はBecause文を導入したJasonエージェント、その他はサンプルエージェントである。また、エージェント番号6, 10は死亡している。Agent[07]を含む複数のエージェントがBecause文を用いてAgent[03] (狂人)を怪しむ発言をしている。霊媒師であるAgent[02]は、対抗COをした偽霊媒師のAgent[15] (人狼)を怪しむ発言をしている。しかし、他のエージェントの発言を聞き、投票先をAgent[15]からAgent[03]に変更している。これにより、他者と協力して怪しいと思われるプレイヤーを確実に吊る行動を実現している。

```
2, status, 1, BODYGUARD, ALIVE, Jason005
2, status, 2, MEDIUM, ALIVE, Jason004
2, status, 3, POSSESSED, ALIVE, Sample09
2, status, 4, VILLAGER, ALIVE, Jason002
2, status, 5, VILLAGER, ALIVE, Sample08
2, status, 6, SEER, DEAD, Jason001
2, status, 7, VILLAGER, ALIVE, Sample07
2, status, 8, WEREWOLF, ALIVE, Sample06
2, status, 9, VILLAGER, ALIVE, Sample12
2, status, 10, WEREWOLF, DEAD, Sample11
2, status, 11, VILLAGER, ALIVE, Sample10
2, status, 12, VILLAGER, ALIVE, Sample05
2, status, 13, VILLAGER, ALIVE, Sample15
2, status, 14, VILLAGER, ALIVE, Sample14
2, status, 15, WEREWOLF, ALIVE, Sample13
2, talk, 0, 0, 14, BECAUSE (AND (ANY ATTACKED Agent[06]) (DAY 1 (Agent[03] DIVINED Agent[06] WEREWOLF))) (XOR (ESTIMATE Agent[03] WEREWOLF) (ESTIMATE Agent[03] POSSESSED))
2, talk, 2, 0, 5, BECAUSE (AND (ANY ATTACKED Agent[06]) (DAY 1 (Agent[03] DIVINED Agent[06] WEREWOLF))) (XOR (ESTIMATE Agent[03] WEREWOLF) (ESTIMATE Agent[03] POSSESSED))
2, talk, 3, 0, 3, DAY 2 (DIVINED Agent[05] HUMAN)
2, talk, 4, 0, 2, COMINGOUT Agent[02] MEDIUM
2, talk, 5, 0, 8, BECAUSE (AND (NOT (COMINGOUT Agent[03] WEREWOLF)) (NOT (COMINGOUT Agent[06] WEREWOLF))) (DAY 1 (Agent[03] DIVINED Agent[06] WEREWOLF))) (ESTIMATE Agent[03] POSSESSED)
2, talk, 6, 0, 11, BECAUSE (AND (ANY ATTACKED Agent[06]) (DAY 1 (Agent[03] DIVINED Agent[06] WEREWOLF))) (XOR (ESTIMATE Agent[03] WEREWOLF) (ESTIMATE Agent[03] POSSESSED))
2, talk, 7, 0, 7, BECAUSE (AND (ANY ATTACKED Agent[06]) (DAY 1 (Agent[03] DIVINED Agent[06] WEREWOLF))) (XOR (ESTIMATE Agent[03] WEREWOLF) (ESTIMATE Agent[03] POSSESSED))
2, talk, 9, 0, 12, BECAUSE (AND (ANY ATTACKED Agent[06]) (DAY 1 (Agent[03] DIVINED Agent[06] WEREWOLF))) (XOR (ESTIMATE Agent[03] WEREWOLF) (ESTIMATE Agent[03] POSSESSED))
2, talk, 11, 0, 15, COMINGOUT Agent[15] MEDIUM
2, talk, 12, 0, 13, BECAUSE (AND (ANY ATTACKED Agent[06]) (DAY 1 (Agent[03] DIVINED Agent[06] WEREWOLF))) (XOR (ESTIMATE Agent[03] WEREWOLF) (ESTIMATE Agent[03] POSSESSED))
2, talk, 14, 1, 15, BECAUSE (AND (NOT (COMINGOUT Agent[03] WEREWOLF)) (NOT (COMINGOUT Agent[06] WEREWOLF))) (DAY 1 (Agent[03] DIVINED Agent[06] WEREWOLF))) (ESTIMATE Agent[03] POSSESSED)
2, talk, 16, 1, 2, DAY 2 (IDENTIFIED Agent[10] WEREWOLF)
2, talk, 26, 2, 13, BECAUSE (XOR (ESTIMATE Agent[03] WEREWOLF) (ESTIMATE Agent[03] POSSESSED)) (AND (VOTE Agent[03]) (REQUEST ANY (VOTE Agent[03])))
2, talk, 27, 2, 3, AND (VOTE Agent[01]) (REQUEST ANY (VOTE Agent[01]))
2, talk, 28, 2, 5, BECAUSE (XOR (ESTIMATE Agent[03] WEREWOLF) (ESTIMATE Agent[03] POSSESSED)) (AND (VOTE Agent[03]) (REQUEST ANY (VOTE Agent[03])))
2, talk, 29, 2, 7, BECAUSE (XOR (ESTIMATE Agent[03] WEREWOLF) (ESTIMATE Agent[03] POSSESSED)) (AND (VOTE Agent[03]) (REQUEST ANY (VOTE Agent[03])))
2, talk, 30, 2, 8, BECAUSE (ESTIMATE Agent[03] POSSESSED) (AND (VOTE Agent[03]) (REQUEST ANY (VOTE Agent[03])))
2, talk, 31, 2, 1, AND (VOTE Agent[03]) (REQUEST ANY (VOTE Agent[03]))
2, talk, 32, 2, 12, BECAUSE (XOR (ESTIMATE Agent[03] WEREWOLF) (ESTIMATE Agent[03] POSSESSED)) (AND (VOTE Agent[03]) (REQUEST ANY (VOTE Agent[03])))
2, talk, 33, 2, 9, AND (VOTE Agent[03]) (REQUEST ANY (VOTE Agent[03]))
2, talk, 34, 2, 11, BECAUSE (XOR (ESTIMATE Agent[03] WEREWOLF) (ESTIMATE Agent[03] POSSESSED)) (AND (VOTE Agent[03]) (REQUEST ANY (VOTE Agent[03])))
2, talk, 35, 2, 4, AND (VOTE Agent[03]) (REQUEST ANY (VOTE Agent[03]))
2, talk, 36, 2, 2, BECAUSE (AND (COMINGOUT Agent[02] MEDIUM) (Agent[15] COMINGOUT Agent[15] MEDIUM)) (XOR (ESTIMATE Agent[15] WEREWOLF) (ESTIMATE Agent[15] POSSESSED))
2, talk, 37, 2, 14, BECAUSE (XOR (ESTIMATE Agent[03] WEREWOLF) (ESTIMATE Agent[03] POSSESSED)) (AND (VOTE Agent[03]) (REQUEST ANY (VOTE Agent[03])))
2, talk, 38, 2, 15, DAY 2 (IDENTIFIED Agent[10] HUMAN)
2, talk, 42, 3, 15, AND (VOTE Agent[02]) (REQUEST ANY (VOTE Agent[02]))
2, talk, 44, 3, 2, BECAUSE (AND (COMINGOUT Agent[02] MEDIUM) (Agent[15] COMINGOUT Agent[15] MEDIUM)) (XOR (ESTIMATE Agent[15] WEREWOLF) (ESTIMATE Agent[15] POSSESSED))
2, talk, 63, 4, 2, AND (VOTE Agent[03]) (REQUEST ANY (VOTE Agent[03]))
2, vote, 1, 3
2, vote, 2, 3
2, vote, 3, 1
2, vote, 4, 3
2, vote, 5, 3
2, vote, 7, 3
2, vote, 8, 3
2, vote, 9, 3
2, vote, 11, 3
2, vote, 12, 3
2, vote, 13, 3
2, vote, 14, 3
2, vote, 15, 2
2, execute, 3, POSSESSED
```

Agent[07]がAgent[03]は人狼か狂人であると発言

Agent[02]がAgent[15]を怪しむ発言

Agent[02]が、Agent[07]を信じ、投票先をAgent[03]に変更

図4 実行結果のログ

次に、エージェントの思考部分である Jason の出力結果を示す。これは、図 4 の例に出てきた霊媒師 (Agent[02]) の思考結果である。Because 文を用いたエージェントと対象者のリストが送られてきている。そして、発言者の投票度を基に昇順に並び変えた結果、発言者の投票度が最小なのは Agent[07] であった。また、発言者 (Agent[07]) の投票度が対象者 (Agent[03]) の投票度より小さいので、今回は Agent[07] を信頼することに決めた。その結果、投票度から決めた投票先 (Agent[15]) から、Because 文から判断した投票先 (Agent[03]) へ変更された。実際の Jason の実行結果を図 5, 6 で示す。図 6 は出力結果の中から一部を抜粋したものである。

```
[clientD] From server: [vote, [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15], [1, 3, 4, 5, 7, 8, 9, 11, 12, 13, 14, 15], [[6, seer], [3, seer], [10, seer], [15, medium], [2, medium]], [[2, 6], [5, 3]], [[9, 10], [6, 3]], [[10, 15]], [15, [6]], [[19, 3], [6, 3], [7, 3], [8, 3], [12, 3], [11, 3], [14, 3]], [], []]
[clientD] 13, 3
[clientD] [[-0.13999999999999999, 13, 0.66, 3]]
[clientD] 5, 3
[clientD] [[-0.13999999999999999, 13, 0.66, 3], [0.030000000000000006, 5, 0.66, 3]]
[clientD] 7, 3
[clientD] [[-0.20000000000000004, 7, 0.66, 3], [-0.13999999999999999, 13, 0.66, 3], [0.030000000000000006, 5, 0.66, 3]]
[clientD] 8, 3
[clientD] [[-0.20000000000000004, 7, 0.66, 3], [-0.13999999999999999, 13, 0.66, 3], [0.030000000000000006, 5, 0.66, 3], [0.13999999999999999, 8, 0.66, 3]]
[clientD] 15, 3
[clientD] [[-0.20000000000000004, 7, 0.66, 3], [-0.13999999999999999, 13, 0.66, 3], [0.030000000000000006, 5, 0.66, 3], [0.09, 12, 0.66, 3], [0.13999999999999999, 8, 0.66, 3]]
[clientD] 11, 3
[clientD] [[-0.20000000000000004, 7, 0.66, 3], [-0.18000000000000002, 11, 0.66, 3], [-0.13999999999999999, 13, 0.66, 3], [0.030000000000000006, 5, 0.66, 3], [0.09, 12, 0.66, 3], [0.13999999999999999, 8, 0.66, 3]]
[clientD] 14, 3
[clientD] [[-0.20000000000000004, 7, 0.66, 3], [-0.18000000000000002, 11, 0.66, 3], [-0.17, 14, 0.66, 3], [-0.13999999999999999, 13, 0.66, 3], [0.030000000000000006, 5, 0.66, 3], [0.09, 12, 0.66, 3], [0.13999999999999999, 8, 0.66, 3]]
[clientD] 1, were[0]posessed[0.05]seer[0.9]
[clientD] 1, villager vote[-0.7350000000000001]
[clientD] 3, were[0.2]posessed[0.0]seer[0]
[clientD] 3, villager vote[0.66]
[clientD] 4, were[0]posessed[0]seer[0]
[clientD] 4, villager vote[0]
[clientD] 5, were[0]posessed[0]seer[0]
[clientD] 5, villager vote[0]
[clientD] 7, were[0]posessed[0]seer[0]
[clientD] 7, villager vote[0]
[clientD] 8, were[0]posessed[0]seer[0]
[clientD] 8, villager vote[0]
[clientD] 9, were[0]posessed[0]seer[0]
[clientD] 9, villager vote[0]
[clientD] 11, were[0]posessed[0]seer[0]
[clientD] 11, villager vote[0]
[clientD] 12, were[0]posessed[0]seer[0]
[clientD] 12, villager vote[0]
[clientD] 13, were[0]posessed[0]seer[0]
[clientD] 13, villager vote[0]
[clientD] 14, were[0]posessed[0]seer[0]
[clientD] 14, villager vote[0]
[clientD] 15, were[0.3]posessed[0.2]seer[0]
[clientD] 15, villager vote[0.8400000000000001]
[clientD] [[Candidate, [1, 3, 4, 5, 7, 8, 9, 11, 12, 13, 14, 15]]
[clientD] ["Agent", 1, "villager vote", -0.7350000000000001]
[clientD] ["Agent", 3, "villager vote", 0.66]
[clientD] ["Agent", 1, "villager vote", -0.7350000000000001]
[clientD] ["Agent", 3, "villager vote", 0.66]
[clientD] ["Agent", 3, "villager vote", 0.66]
[clientD] ["Out", [15, 3, 9, 12, 8, 11, 14, 5, 7, 13, 4, 1]]
[clientD] Agent7 is suspicious of Agent3
[clientD] I believe Agent7
[clientD] vote target is changed! 15 -> 3
[clientD] To server: 3
```

図 5 Jason の出力結果

```
[clientD] From server: [vote, [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15], [1, 3, 4, 5, 7, 8, 9, 11, 12, 13, 14, 15],
[[6, seer], [3, seer], [10, seer], [15, medium], [2, medium]], [[2, 6], [5, 3]], [[9, 10], [6, 3]], [[10, 15]], [15, [6]],
[[13, 3], [5, 3], [7, 3], [8, 3], [12, 3], [11, 3], [14, 3]], [], []]
```

Because文を用いた発言者と対象者のリスト
(複数のエージェントがBecause文を用いてAgent[03]を疑う発言をしている)

```
[clientD] [[-0.20000000000000004, 7, 0.66, 3], [-0.18000000000000002, 11, 0.66, 3], [-0.17, 14, 0.66, 3],
[-0.13999999999999999, 13, 0.66, 3], [0.030000000000000006, 5, 0.66, 3], [0.09, 12, 0.66, 3], [0.13999999999999999, 8, 0.66, 3]]
```

Agent[07]の投票度(-0.2) < Agent[03]の投票度(0.66)

```
[clientD] ["Out", [15, 3, 9, 12, 8, 11, 14, 5, 7, 13, 4, 1]]
[clientD] Agent7 is suspicious of Agent3
[clientD] I believe Agent7
[clientD] vote target is changed! 15 -> 3
[clientD] To server: 3
```

Agent[15]に投票する予定

Agent[07]を信頼し、投票先をAgent[15]からAgent[03]に変更

図 6 Jason の出力結果 (一部抜粋)

表 2 実行結果

村人陣営	人狼陣営	村人勝率
Because 文 導入 Jason	サンプルエージェント	10.3%
Because 文 未導入 Jason	サンプルエージェント	14.3%

4.2 実験結果

Because 文の導入がエージェントの強さにどのような影響を与えたかを検証するために、Because 文を導入した Jason エージェントを 4 体使用した場合と、導入していない Jason エージェントを 4 体使用した場合で、勝率を比較した。4 体の Jason エージェントの役職は占い師、霊媒師、狩人、村人に固定した。今回の実験では、怪しいエージェントの Because 文に騙されず、村人陣営側の要請に従えるかが重要であるため、人狼陣営側のエージェントは Because 文を発言できる必要がある。しかし、人狼知能大会出場エージェントの中に、Because 文を発言するエージェントは見受けられなかった。そのため、残りの村人及び人狼陣営は Because 文を発言できるサンプルエージェントを用いた。15 人によるゲームを各条件で 300 回ずつ行った結果、Because 文の導入により勝率が下がった。勝率が下がった原因として、発言の信頼度をエージェントの怪しさだけから決定したことが考えられる。現状では、発言者、対象者の怪しさをもとに行動の選択を行っているため、役職推定の値が動かない場合、正しく信頼度を測ることができなかったと思われる。また、投票度の変更を行う際のルールがまだ十分に適切でないことも問題であると思われる。今回は、他のエージェントの投票先によって常に投票度を変化させている。しかし、現状では偶然自分と投票先が一致した人の投票度も下げてしまっているため、自分の Because 文によって投票先を変えた人のみ投票度を下げるなどの制約をつけるべきであったと考えられる。

5 まとめ

本研究では、人狼エージェントの対戦を人間同士の対戦に近づける一環として、他者を説得させ協力プレイを可能にすることを目指し、Jason による人狼エージェントに Because 文を利用する機能を導入した。Because 文を発言し、発言された Because 文を判別することによって他者の行動を変化させることが可能になった。しかし発言者や対象者の信頼度から発言の信頼性を測ったために、役職推定のパラメータが動いていない際には信頼度を正確に測ることができなかった。今後の課題として、発言内容のみから信頼性を判断することのできるエージェントの作成が挙げられる。発言のみから信頼度を測るには信頼できる発言と信頼できない発言の 2 種類を発言可能なエージェントが必要である。しかし、現時点での人狼エージェントは矛盾した発言は行わない。そのために、まずは矛盾した発言をする人狼エージェントの作成が必要である。そして、その発言の矛盾を検知する方法の確立が必要であると考えられる。また、村人陣営側と同等に戦わせるために、人狼陣営側の役職推定も必要である。これらの課題を解決することによって、より人間らしい行動を行うエージェントに近づくと考えられる。また、矛盾を検知する思考過程を出力することによって、ゲーム後のストーリーが面白いものになることも期待できる。そして、これらの課題をクリアしてより妥当な戦略を取れるために必要な能力の獲得として、今回の Because 文を利用する機能の導入はそれ自体有意義であったと考える。その他の課題として、能力者の CO タイミングや占い先の決定方法などの基本的戦略の強化や、その他人数ルールへの対応が挙げられる。

謝辞

本研究を進めるにあたり、最後まで丁寧かつ熱心に指導して下さっ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, 片上大輔. 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, 片上大輔, 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. *The 19th Game Programming Workshop 2014*, pp. 157-161, 2014.
- [3] 新出尚之, 高田司郎. 不確実な信念を持つエージェントの確率的戦略とBDIモデル. *The 30th Annual Conference of the Japanese Society for Artificial Intelligence*, 2016.
- [4] 池内加奈. 人狼ゲームをプレイするエージェントの実装. 2016年度卒業論文, 奈良女子大学理学部情報科学科, 2017.
- [5] 鳥海不二夫, 稲葉通将, 大澤博隆, 片上大輔, 松原仁, 狩野芳伸, 大槻恭士, Claus Aranha, 伊藤毅志. 人狼知能プロジェクト. <http://aiwolf.org>.
- [6] Rafael H. Bordini and Michael Wooldridge and Jomi Fred Hübner. *Programming multi-agent systems in AgentSpeak using Jason*. John Wiley & Sons, 2007.
- [7] 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>.
- [8] 人狼知能知能プロトコルの仕様 (ver3.6, 2019年度版). http://aiwolf.org/control-panel/wp-content/uploads/2019/02/protocol_2019_3_6m.pdf.
- [9] 田中里穂. 論理的思考が行える人狼プレイヤーエージェントの構築. 2019年度卒業論文, 奈良女子大学生活環境学部情報衣環境学科生活情報通信科学コース, 2020.