

卒業論文

多人数ゲームにおける交渉選択手法の提案

平成 26 年 2 月 7 日提出

指導教員 近山 隆 教授
鶴岡 慶雅 准教授

電子情報工学科

03120394 伊藤 義章

目次

1	はじめに	1
1.1	背景	1
1.2	目的	1
1.3	本研究の貢献	2
1.4	本稿の構成	2
2	関連研究	2
2.1	多人数ゲーム	2
2.2	交渉	2
2.3	UCT アルゴリズム	3
2.4	多人数ゲームにおける UCT アルゴリズム	4
3	提案手法	4
3.1	交渉の利益見積り	5
3.2	交渉を選ぶ評価関数	6
4	実験方法	7
4.1	カタンの開拓者	7
4.2	SmartSettlers	10
4.3	問題の簡単化	10
4.4	交渉の方針	11
5	評価	12
5.1	交渉を行なわない場合の比較	12
5.2	交渉の利益計算方法の比較	13
5.3	交渉の評価基準の比較	16
6	おわりに	20
6.1	まとめ	20
6.2	今後の課題	20
	謝辞	21
	参考文献	22

概要

実世界では多くの交渉が行われており、競争的状況において自分の利益を最大化することを目指している。交渉の研究の多くは、定性的な研究であり、定量的な交渉の研究は未だ発展途上である。その理由として、先読みによって得られる利益を数値化し、どのような交渉を提示するか指標となる評価関数を用意する難しさにある。本研究では、先読みによる利益計算に、近年成功をおさめているモンテカルロ法を適用することを提案し、また、指標となる様々な評価関数を提示する。モンテカルロ法が利益計算において有効な手段であるかどうかを検証し、また、評価関数の比較実験を行なう事で交渉において重要となる要素を検証することを目的とする。実験結果より、モンテカルロ法は利益計算に有効であるとは言えなかった。しかし、交渉の評価関数を生成する上で、「提案成功率」「受諾率」「自分の利益」のみならず、「相手の損益」を考慮した評価関数を生成する必要があることが分かった。

1 はじめに

1.1 背景

現実社会では、多くの場面で関係者間に利害関係があり、その解決のために様々な交渉が行われている。交渉の研究は社会科学的な研究が多く、戦略やノウハウといった定性的な交渉術 [1, 2, 3] としての研究が中心となっている。そのため、交渉において定量的な交渉術の研究は未だ発展途上であり [4]、決定的な解は見つからない。

決定的な解が見つからない理由として、自分が最も有利となる交渉案を定量的に選択する難しさにある。ひとつ目は、交渉によって得られる利益を数値化することの難しさである。正確に利益を数値化するには、現在の状態を考えるだけでは不十分であり、不確実な将来に得られる利益を含めるのは困難である。ふたつ目は、相手が交渉を受ける可能性を、定量的に評価する難しさである。もし、正確な先読みによって利益計算を行い、自分にとって最も利益の高い交渉を相手に提示したとしても、相手が受諾しなければ、自分は利益を得ることができない。そのため、相手が交渉をうける可能性を考慮し、自分の利益の期待値を最大にする交渉案を生成する評価関数が必要である。

また、交渉を行う人工知能を、実世界の条件のままコンピュータ上に設計する事は困難である。なぜなら、実世界の問題では、とりうる状態が無数に存在し、各々から遷移しうる先の状態がまた無数にあり、遷移が可能になる条件も複雑である。よって、実世界をモデル化するために、実世界の特徴となる要素を抽出し設計する必要がある。

要素を抽出した知能ゲームをコンピュータ上に設計しプレイさせて、実世界に応用する研究 [5] が以前から行われている。一般に実世界をモデル化したゲームの要素を、多人数ゲームであるか、確率的な要素を含むか、ある状態の情報がわかっているか、という軸でわけることが多い。多人数でプレイしたり、不確定あるいは不完全な情報しか得られないゲームが、最も実世界に近いゲームである。しかし、多人数ゲームはプレイヤー同士の協力関係を持つと戦略を変える必要があり、また、不確定不完全情報ゲームは、偶発的なイベントで状況が変動するため探索空間が大きくなるため純粋な先読みが難しい。本研究では、この知能ゲームとして「カタンの開拓者」(以下カタン)というゲームを用いて評価検証を行なった。

1.2 目的

自らの利益の期待値を最大にする交渉を定量的に検証するために、利益を数値化し、評価関数を用いて交渉を選択する必要がある。本研究では、先読みによる利益計算に、ランダム試行に基づくモンテカルロ法を適用し、その有効性を検証することを目的とする。モンテカルロ法は人工知能の分野で多くの成功をおさめているため、利益計算に応用できると期待される。また、交渉の選択を行なう評価関数として、様々な評価関数の

比較実験を行なう事で交渉において重要となる要素を検証することを目的とする。

1.3 本研究の貢献

利益計算において、簡単なルールベースとモンテカルロ法の一致率を検証することで、モンテカルロ法による利益計算の問題点を明らかにした。また、交渉案を選択する評価関数を生成する時に、悪い交渉案を混ぜることが有意な結果を残したことから、「相手の損益」を考慮した評価関数が有効であると考えられる。本研究によって得られたゲームモデルにおける交渉案選択手法は、他の実世界に近いゲームモデルで応用することが可能である。

1.4 本稿の構成

本稿は、第2章で関連研究の紹介、第3章で提案手法を説明し、第4章で実験方法、第5章で評価結果を示し、最後に第6章でまとめと今後の課題について述べる。

2 関連研究

本章では、多人数ゲームシステムの説明を2.1節で行い、本研究のメインテーマである交渉に関する説明を2.2節で行う。

次に2.3節、2.4節において、交渉の評価値を計算するUCTアルゴリズムについて説明する。

2.1 多人数ゲーム

自ら考え行動し、人間に変わって複雑な作業などを代行するものを「エージェント」と呼ぶ。そのようなエージェントが多数共存する環境を、多人数ゲームと呼び、複数のエージェントが協調して解決しようとするシステムがマルチエージェントシステムである。社会にはいくつもの要素が絡み合う複雑な事象が溢れており、個々のエージェントの動きだけでは想定できないようなマクロな現象をボトムアップ的に解析することが可能となる。

2.2 交渉

2.2.1節で一般的に行われている交渉のパターンを紹介し、2.2.2節で交渉の手順について述べる。最後に2.2.3節で交渉のアルゴリズムを紹介する。

2.2.1 協力型交渉と競争型交渉

交渉には大きく協力型交渉と競争型交渉がある [6]。一般に、競争型交渉は協力型交渉に比べて難しい。なぜなら、協力型交渉はお互いの情報が開示されている場合がほとんどで、お互いの利益を最大化するだけでよい。そのため、交渉は成立しやすい。一方で、競争型交渉は多くの場合お互いの情報が開示されておらず、また、各々のエージェントが自分の利益のみ最大化することを考えるため、交渉は決裂することが多い。そのため、妥協点をいかにうまく設定するかが、現実世界でも大きな課題である。

また、一般的に競争型交渉は価格交渉のようにゼロサムゲームを想定し、最初の交換提案をどの程度に、それぞれがどのように譲歩するかを決定する。しかし、交渉によっては交換する資源の組み合わせを変更する事で、双方の利益をふやすことができる場合があり、単なるゼロサムゲームでないゲームもある。このような交渉形態は、利益交換型交渉 [7] と呼ばれ、ゼロサムゲームである分配型交渉よりも良い交渉形態である。この交渉を扱うことができることも、カタンで交渉の研究を行なう利点となる。

2.2.2 交渉の手順

交渉において各々のプレイヤーは以下のような手順で交渉の成立を目指す [4]。

候補の列挙

基本的に、交渉を行うときは、他のプレイヤー全てと交渉することが可能である。全てのプレイヤーに対して、1 対 1、1 対多、多対多の交渉内容を列挙する必要がある。

候補の選択

列挙した候補の中から、交渉を選択する基準（ゲームの場合は、評価関数）に基づいて評価し選択する必要がある。それぞれの候補に対して、交渉を行った事によって、お互いの利益の増加がおこりえるが、相手と自分の利益が同程度上昇しない限りは基本的に交渉は成立しない。それぞれの交渉に対して「正確な利益を計算し、利益に基づいた良い選択基準」を作る事が交渉を成立させる重要な要因である。また、交渉相手が消極的なエージェントであった場合、自分にとって少し有利な条件を選択する評価関数に変更することができる。

交渉の切り出し

評価関数に基づいて選択した候補の中から、そのうち上位のいくつかを選択して提案する。これは、現実世界において同じ相手に対し、全ての交渉案を提案することはほとんどないためである。交渉内容をある程度しぼって提案する必要がある。これらが実際に相手に提案する交渉案となる。また、相手に提示する交渉案の順番によって、交渉の成功確率が変化することがある。

交渉の成立

相手が自分の提案した交渉内容を受諾した場合、交渉成立となる。また、提示した交渉案が受諾される事なく、相手によって新たな条件で提案される場合は、相手が提案してきた交渉内容が自らの評価関数に基づいた交渉順位のどこに位置するかをみることで、お互いが受け入れるラインを再度提案する。一定の提案回数で交渉が成立しない場合は、交渉決裂となる。

2.2.3 交渉のアルゴリズム

2.3 UCT アルゴリズム

まずモンテカルロ法について説明する。モンテカルロ法は、状態の評価が難しい問題において有効とされる手法である。

モンテカルロ法は、ある局面から終局までランダムに行われる試行に基づき統計的に評価する。これはプレイアウトと呼ばれ、プレイアウトを繰り返す事で、ある局面の勝率を計算することができる。最も単純なモンテカルロ法は、ある局面の合法手を行ったところからプレイアウトを開始して、毎ターン勝率を計算し、最も勝率の高いものを選択するものである。

しかしながら、有限な時間内に正確な勝率評価を行う場合、明らかに有望と思われる手に対して、有望な手と同様の試行回数を行うことは無益である。よって、これを解消するため有望な手にバイアスをかけて試行回数を配分する必要がある。これが UCB アルゴリズム (Upper Confidence Bound、以下 UCB) である。

バイアスをかけて子ノードを選択する場合、まず未探索の子ノードを優先的に選択する。子ノードが全て探索済みの場合は、以下の式が最大となる子ノードを探索することで、探索回数が少なく、高い報酬を得る可能性のある手に対して第 2 項が大きくなるため探索されやすいというアルゴリズムである。探索回数が多くなると、第 1 項の平均報酬の高い手が基本的に選択される。

$$UCB(j) = \bar{X}_j + C \sqrt{\frac{\ln n}{n_j}} \quad (1)$$

但し、 \overline{X}_j は現在の状態までの試行で算出した平均勝率、 C は調整パラメータ、 n は全ての合法手の試行回数の合計、 n_j は合法手 j の試行回数とする。

選択したノードをプレイアウトした後は、その結果を報酬として探索経路上のノードを更新する。

以上の「子ノードの選択」「子ノードの展開」「プレイアウト」「結果の更新」を終了条件（時間や探索回数など）となるまで繰り返し、平均報酬の最も高い子ノードを次のアクションとする。本研究では、モンテカルロ木探索に UCB 値を用いた探索である UCT アルゴリズム [8, 9] によって得られた子ノードの平均報酬の結果を用いて、交渉の評価関数を生成することとなる。

また、モンテカルロ法のアプローチであるモンテカルロ木探索 (Monte Carlo Tree Search、以下 MCTS) をカタンの AI である JSettlers に適用し、成功をおさめている [10]。JSettlers のプレイヤーに対して、JSettlers をベースとした MCTS の 0 回シミュレート、1,000 回シミュレートおよび 10,000 回シミュレートを行った。1 人の SmartSettlers プレイヤー、他の 3 人 JSettlers を用いたプレイヤーに対し、25%、27%、49% の勝率を残し [?], カタンにおける MCTS がランダムプレイヤーよりは強いことを示している。

2.4 多人数ゲームにおける UCT アルゴリズム

多人数のゲームでは 2 人のゲームに比べて UCT よりも評価関数による局面評価が難しい。[11]

図 2 のように、各ノードが自分のノードの値を最大とする手を選んでいくものとする。ここで、敵となるプレイヤーが 2 人いるため、敵がどちらを選択すべきか判断するためには 2 人の戦略モデルを知る必要がある。しかし、一般に適切な戦略モデルを作成することは難しい。一方で、これに UCT アルゴリズムを適用するとノード B2 ではプレイヤー B が返す報酬は同じであるため、UCB 値による手の選択は探索回数によって決定される。結果、両方の子ノードは同じ回数探索が行われ、親のノード 2 の評価値は 2 つのノードの平均値をとることになる。これにより、他のプレイヤーに対しての混合戦略 [12] を得る事ができる。

以上より、UCT アルゴリズムを多人数ゲームで使用すると、シミュレーションに基づいた探索が行われるため、相手によって戦略が異なる場合がでてくる。そのため、これらの戦略をうまく組み合わせた戦略（混合戦略）を考える必要がある。実際、UCT アルゴリズムは評価関数を用いた Maximum 探索よりも高い性能を発揮している。

3 提案手法

本研究では、現実世界のモデルに近い、不完全情報ゲーム、非決定的、多人数ゲームという要素をもつことから、カタンの開拓者を用いて検証する。交渉を行なうときに以下の手順を踏むものとする。

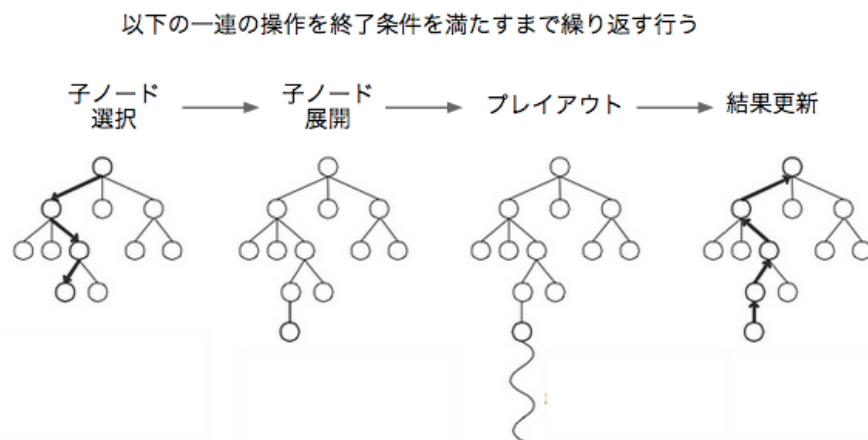


図 1. モンテカルロ木探索 [8]

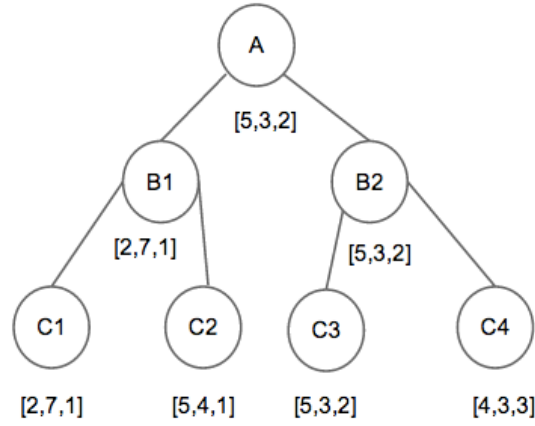


図 2. 多人数ゲームの評価値例

- 全てのプレイヤーに対して可能な交渉案のリストアップ
- 各交渉案の利益の見積もり
- 評価関数による各交渉案の期待値計算
- 交渉案の提示

自分のターンにおいて、初めに、全てのプレイヤーに対して可能な交渉案を求める。次に、これらの交渉案に対して、ルールベースもしくは UCT アルゴリズムを用いて、利益の見積もりを行なう。そして、見積もりの結果を用いて、提案した評価関数に基づき、交渉による各交渉案の期待値を計算する。最も自分の利益の高い交渉案を提示せずに、評価関数を用いて期待値を計算する必要があるのは、相手が受諾するかどうかを考慮する必要があるからである。最後に、最も期待値の高い交渉案を提示する。交渉は自分のターンに 1 回限りとする。

以下、本章では、まず交渉の利益計算方法について 3.1 節で説明し、3.2 節で交渉を選ぶ評価関数を説明する。

3.1 交渉の利益見積り

交渉を提案する上で自分と相手の利益を計算する必要がある。利益を以下のように計算するものとする。

利益計算において、先読み手法である UCT アルゴリズムを用いることが本研究の提案の 1 つである。UCT アルゴリズムによるプレイアウトによって、各プレイヤーの勝率を計算することができる。交渉による損得は最終的に自分の勝率に直結することから、UCT アルゴリズムの先読みによって予め得た勝率により利益計算を行なうことができると考えられる。よって、UCT アルゴリズムの先読みによって得た勝率を、利益計算に用いる事にする。まず、交渉を行わなかった場合の勝率を x 、交渉を行って成功した場合の勝率を x_{trade} とする。

$$R = x_{trade} - x \quad (2)$$

「利益 (R) = 交渉をおこなった場合と行わなかった場合とを比べて、自分の勝率が何ポイント上昇するか」

「それぞれの交渉を行って成功した場合」と「交渉を行わなかった場合」について勝率の違いを求める。

- 交渉を行わなか
 - プレイアウトした時の自分プレイヤー A の勝率 W_a
 - プレイアウトした時の相手プレイヤー B の勝率 W_b
- 交渉を行なって成功した

- プレイアウトした時の自分プレイヤー A の勝率 Wt_a
- プレイアウトした時の相手プレイヤー B の勝率 Wt_b

すなわち各々のプレイヤーの利益は以下ようになる。

$$R_a = Wt_a - W_a \quad (3)$$

$$R_b = Wt_b - W_b \quad (4)$$

3.2 交渉を選ぶ評価関数

相手のプレイヤーが自分の提案した交渉を必ず応じてくれることはなく、自分の利益を最大化しつつ、相手が受諾しやすい交渉案を提示する必要がある。プレイヤー間では競争的交渉が行われるため、その中で妥協点をみつけ出す必要がでてくる。すなわち、妥協的交渉を行う方法を考える必要がある。それぞれの交渉に関して評価値を設定する事により、評価値が最も高くなる交渉を提案することにする。ただし、全ての交渉案に対し、交渉をしない方が評価値が高い場合は交渉を行なわないものとする。

$$v_{best} = \max_i f(s_i) (s = 1, 2, \dots, N) \quad (5)$$

「最大の評価値 (v_{best}) = 最も高い評価値の交渉案」「評価関数 ($f(s)$) = ある交渉案に対して評価値を求める関数」「交渉案 (s) = 提案することのできる交渉案」

本研究では、前節で得られた利益計算の結果を用いて、交渉の成功率を考慮した評価関数により交渉を提示する。自分の利益を最大化しつつ、交渉の成功率をあげる必要がある。交渉の成功率は「相手が得られる利益」に影響を受けるものと仮定する。よって、良い交渉案を選ぶ評価関数として、以下の5つの評価方法を提案することにする。まず、自分の利益のみしか考えない「自己中心的交渉」を行なうプレイヤーをベースラインとして扱う。次に、交渉の成功率よりも相手の利益を優先にした「利益優先交渉」と交渉の成功率を優先した「受諾優先交渉」を提示する。また、お互いの利益をバランスよく配分することで、自分の利益と交渉の成功率を調整する手法として「和交渉」「積交渉」を提示する。

自己中心的交渉

自己中心的交渉の評価関数は、「自分の利益を最大にする」関数である。相手の利益を全く考慮しないプレイヤーとしてベースラインに用いるものとする。全てのプレイヤーに対して、考えられる全ての交渉案から、交渉を行なうことによって自分の利益が最大となる交渉案を選び、他プレイヤーに提示する。交渉をしない場合、最大の評価値をとる交渉案であった場合、交渉は行なわないものとする。

$$v_{best} = f(s) = \max\{R_a(s) : s = 1, 2, \dots, N\} \quad (6)$$

利益優先交渉

利益優先の評価関数は、「自分の利益を最大にし、相手の利益を最小にする」関数である。まず、全てのプレイヤーに対して、考えられる全ての交渉案から、交渉を行い成功することによって「相手」の利益がプラスとなる交渉案を選ぶ。選択された全ての交渉案の中から、自分の利益を最大にする交渉案を選択し、あるプレイヤーに提示する。交渉をしない場合、最大の評価値をとる交渉案であった場合、交渉は行なわないものとする。

$$v_{best} = f(s) = \max\{R_a(s) : R_b(s) >= 0, s = 1, 2, \dots, N\} \quad (7)$$

受諾優先交渉

自分(プレイヤー A とする)の評価関数と相手(プレイヤー B とする)の評価関数が違う場合、プレイヤー A の評価関数ではプレイヤー B の利益がプラスであったとしても、プレイヤー B の評価関数ではプレイヤー

B の利益がマイナスであると評価する場合がある。交渉において、この評価関数の違いによって受諾、拒否が決まる。もし、交渉案を提示し断られてしまった場合、別の提案であれば得られるはずだった利益の機会損失を生じてしまう。よって、交渉成立を優先とすることで機会損失を最小にする戦略が考えられる。相手との交渉成功率をあげるために、相手の利益を最大にするものとする。交渉成立優先の評価関数は、「自分は必ず利益を得られる中で、相手の利益を最大にする」関数である。まず、全てのプレイヤーに対して、考えられる全ての交渉案から、交渉を行い成功することによって「自分」の利益がプラスとなる交渉案を選ぶ。選択された交渉案の中から、相手の利益を最大にする交渉案を選択し、他プレイヤーに提示する。交渉をしない場合、最大の評価値をとる交渉案であった場合、交渉は行なわないものとする。

$$v_{best} = f(s) = \max\{R_b(s) : R_b(s) > 0, s = 1, 2, \dots, N\} \quad (8)$$

和交渉

和の評価関数は、「自分の利益と相手の利益の合計を最大にする」関数である。まず、全てのプレイヤーに対して、考えられる全ての交渉案から、交渉を行ない成功することによって「自分」の利益がプラスとなる交渉案を選ぶ。選択された全ての交渉案の中から、自分の利益と相手の利益の合計が最大となる交渉案を選択し、他プレイヤーに提示する。交渉を行なわない場合、最大の評価値をとる交渉案であった場合、交渉は行なわないものとする。また、評価値が同じ場合は、ランダムに選択するものとする。

$$v_{best} = f(s) = \max\{R_a(s) + R_b(s) : s = 1, 2, \dots, N\} \quad (9)$$

積交渉

積の評価関数は、「自分の利益と相手の利益の積を最大にする」関数である。まず、全てのプレイヤーに対して、考えられる全ての交渉案から、交渉を行ない成功することによって「自分」の利益がプラスとなる交渉案を選ぶ。選択された全ての交渉案の中から、自分の利益と相手の利益の積が最大となる交渉案を選択し、他プレイヤーに提示する。交渉をしない場合、最大の評価値をとる交渉案であった場合、交渉は行なわないものとする。初めに、自らの利益がプラスになるものを選んでいて、互いの利益がマイナスな交渉案を選択することはない。また、評価値が同じ場合は、ランダムに選択するものとする。

$$v_{best} = f(s) = \max\{R_a(s) * R_b(s) : s = 1, 2, \dots, N\} \quad (10)$$

本研究ではこれらの評価関数を用いて、他のプレイヤーに対して毎ターン上記の交渉を行い、「提案成功率」「受諾率」「勝率」の結果を比較する対戦実験を行なう。

4 実験方法

本章では、まず評価測定を行うカタンの開拓者について 4.1 節で説明し、4.2 節でカタンの AI である SmartSettlers について説明し、4.3 節では複雑なモデルを単純化する。この上で提案手法の利益計算および交渉を行う。最後に、4.4 で交渉を行なうプレイヤーについて説明する。

4.1 カタンの開拓者

本研究でカタンを用いる理由として、1 つ目は実世界のモデルに近いことがあげられる。2 つ目に、現実社会の交渉をモデル化し評価を行う上で、カタンは優れた性質を持っているからである。例えば、ルールによって知識が限定できる点、商交渉における様々な要素（交換、オークション、不完全情報）が含まれている点や、勝敗が明確である事から手段の優劣がわかりやすいという点である。3 つ目に、これまでのカタンの研究対象は、その戦略や先読みを題材にしたものが多く、実際に交渉をおこなうものではなかったからである [8, 13]。

ここでは、4.1.1 節でカタンの概要を説明し、4.1.2 節でカタンのルールを説明する。4.1.3 節でカタンの特徴をのべる。



図 3. カタンの開拓者プレイ中ボードマップ [14]

4.1.1 カタンについて

カタンの開拓者はプレイヤーがゲームボード上からサイコロの出目に応じてリソースを収集する。このリソースを用いて、都市、道路といった開発を行う。ゲームの目的は、少なくとも 10 点を他のプレイヤーよりも先に獲得することである。以下では、カタンのルールをまとめたものである。(但し、多くの詳細を省く)

4.1.2 カタンのルール

ゲームボードと資源

カタンのゲームボードは、6 種類の地形タイルを 19 枚組み合わせて構成されている。組み合わせは、ゲームスタート時にランダムに変更される。地形には「砂漠」「森林(緑)」「丘陵(ベージュ)」「牧草地(黄緑)」「農地(黄色)」「山岳(灰色)」があり、砂漠を除いた残りの地形からはサイコロの出目に応じて資源が得ることができる。得られる資源は、森林から順番に「木」「土」「羊」「麦」「鉄」である。得られた資源を使って、「道路」「開拓地」「街」「発展カード」といった開発を行うことができる。

資源の獲得

各プレイヤーは順に、自分の手番が回ってくる。手番プレイヤーはまずサイコロを 2 つ振る。7 以外の場合は、出た目の合計と同じ数字が書かれた地形から資源が産出することができる(出た目の合計が 7 だった場合は盗賊が出現するが、これについては後述)。サイコロの出目がでた数字の地形に接するように開拓地や街を建設していたプレイヤー全員が、開拓地 1 つ当たり 1 枚(街ならば 2 枚)の対応する資源カードを受け取ることができる。自分の手番でなくても、サイコロの出目がでた地形に自分の開拓地が隣接していれば、資源を受け取ることができる。一方で、自分の手番であっても、サイコロの出目がでた地形に自分の開拓地が隣接していなければ、全く受け取ることはできない。

7 が出た場合、手番のプレイヤーは他のプレイヤーから資源を 1 枚奪うことができる。資源を奪うプレイヤーを指定する必要があるため、他のプレイヤーは「どのプレイヤー」が「どのプレイヤー」から資源を奪ったかを知る事はできる。しかし、奪った資源は他のプレイヤーは知ることはできない。

発展方法

得られた資源を用いて「建設」や「発展カード」を購入することができる。自分のターンの間に、持っている資源で払える限り何回でも購入することができる。それぞれの建設コストは以下のようになっている。

- 道路：「木」「土」を1つずつ
- 開拓地：「木」「土」「羊」「麦」を1つずつ
- 街：「麦」2つと「鉄」3つ
- 発展カード：「羊」「麦」「鉄」を1つずつ

を建設することができる。（建設方法についての詳細な方法はここでは述べない。）

発展カードには5種類あり、発展カードを購入する場合、山札からランダムに1枚獲得することができる。獲得した発展カードは、他のプレイヤーに見せる必要はない。発展カードは自分の手番にのみ使用でき、手番1回ごとに1枚までしか使用することはできない。

資源交換

得られた資源を他の資源に変える方法は全部で3つ存在する。

- 他のプレイヤーとの自由な資源交換
- 港を用いて条件にそった交換
- 同じ資源4つと、任意の資源1つを場から交換

上から順に交渉から得られる利益が多いとされているが、実際は状況や条件によって大きく左右されるため、自分の利益が最大となる交渉を見つける事が、勝利にいち早くつながる方法となる。

また、他プレイヤーと自由な資源交換を行なう場合、交渉内容として考えられる状態数は以下の通りである。プレイヤーAとプレイヤーBが交換するとする。プレイヤーAの手札の枚数 a 枚プレイヤーBの手札の枚数 b 枚とすると、交換する内容は最大で

$$F(a, b) = \sum_i \sum_j a C_i \times b C_j \quad (0 < i < a, 0 < j < b) \quad (11)$$

となる。（ただし、カードの種類の重複がある場合は減る。） $F(a, b)$ 通りの局面が考えられる。

勝利条件

道路や建物を建設する事で、獲得したポイントの合計が最も早く10点に達したプレイヤーが勝利する。得点源は以下のようなものがある。

- 開拓地：1つ当たり1点
- 街：1つ当たり2点
- 得点カード：1枚当たり1点（発展カードの25枚の中に5枚ある）
- 最も長い道路を作った（最長交易路ボーナス）：2点
- 最も多く騎士カードを使った（最大騎士力ボーナス）：2点（発展カードの中に14枚ある）

4.1.3 カタンの特徴

カタンには「マルチエージェント」「非決定的」「不完全情報」という大きく3つの特徴がある。

マルチエージェント

カタンは一般的に4人で行われるゲームである。各々のプレイヤーは自ら考え自立的に行動し、複雑なアクションを行わなければならない。お互いのアクションはこのゲーム上で影響を及ぼし合い、またこのゲーム上で完結している。

非決定的

非決定的な状況は主に2つからおきる。1つ目は「サイコロを振る」ときである。考えられる状況は11通りしかないが、それぞれの状況はランダムに起こりうるため、全ての状況について考える必要がある。また、それぞれの状況がおこる確率はバラバラであるため、確率の低いところでは探索回数を減らし、確率の高いところでは探索回数を増やすといったように、確率に応じた探索方法を提案する必要がある。2つ目は「発展カード」を引くときである。発展カードは25枚の中からランダムに引かれるため、サイコロ同様考えられる状態数が爆発してしまう可能性がある。

不完全情報

相手の情報とは大きく2つあり、「発展カード」と「手札」である。「交渉を行う」場合に不完全情報だと考えられる状態数が急激に増えてしまう。それぞれ完全に知る事ができない理由は1つずつ存在する。発展カードは、山札からランダムにカードが引かれ、引いたプレイヤーのみカードを知る事ができ他のプレイヤーは知る事ができない。手札は、サイコロで7が出た場合に2人のプレイヤー間でカードの移動が行われ、移動したカードの種類は他のプレイヤーは知る事ができない。

4.2 SmartSettlers

SmartSettlers は研究用に開発されたオープンソースのカタン AI である [15]。JSettlers をベースとして開発されており、単位時間あたりのシミュレーション回数を増やす目的で開発された。JSettlers はカタンの AI で、GPL(General Public License) をもつオープンソースプロジェクトである。当初は AI の研究プロジェクトとして作成された [16]。SmartSettlers は交渉を除くカタンの一連の動作を行うことが可能である。SmartSettlers はランダムアクションと UCT アルゴリズムアクションの2つの戦術を有している。エージェントはこれに基づいた行動を行う。いずれもまずは、考えられる選択肢を全て列挙する。ランダムアクションはこれらのアクションの中からランダムに選択し行動する。UCT アルゴリズムアクションはアクションの中から UCT アルゴリズムに基づいて選択し行動する。また、SmartSettlers は研究用に開発されたことから、自己対戦を行なう事が可能である。自己対戦は全てのプレイヤーがランダムにアクションを行なう場合、UCT アルゴリズムでアクションを行なう場合、いずれの場合も可能である。また、このとき「Intel(R) Xeon(R) CPU X5680 @ 3.33GHz」の実験環境のもとで、全ての UCT プレイヤーが1,000回プレイアウトを行なった場合、10数秒で1ゲームを終える事ができる。さらに、カタンは席順によって勝率に差が出る事が一般的に知られている [15]。SmartSettler において、全てのプレイヤーをランダムアクションとし、400ゲームの対戦実験を行なった結果、図4のような結果となった。

また、同様に1,000回シミュレーションを行なう UCT アルゴリズムプレイヤーにおいて400ゲーム行なった結果を以下の図5に示す。

このように席順によって勝率が変わる原因として初期配置が考えられる [15]。席順によって、勝率がかわることから、本研究で提案する手法を用いたアルゴリズムの結果は、席順を入れ替えて全ての席順に対して対戦実験を行い、勝率の平均値をとるものとする。

4.3 問題の簡単化

カタンは「マルチエージェント」「非決定的」「不完全情報」なゲームである。本研究ではこの問題を簡単にするために、コンピュータは不完全情報ゲームよりも完全情報ゲームの方が先読みを得意としていることを利用して、以下のような方法で「マルチエージェント」「非決定的」「完全情報」なゲームであると考えられることにする。理由は以下に示す。

実際のカタンでは相手の情報を完全に知る事はできない。そのため、カタンは不完全情報ゲームである。相手の情報とは大きく2つあり、「発展カード」と「手札」である。

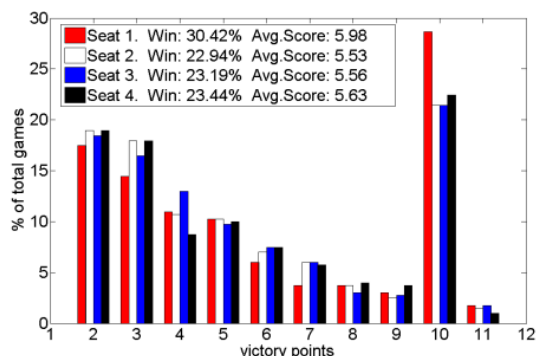


図 4. ランダムアクションによる席順の優劣 [15]

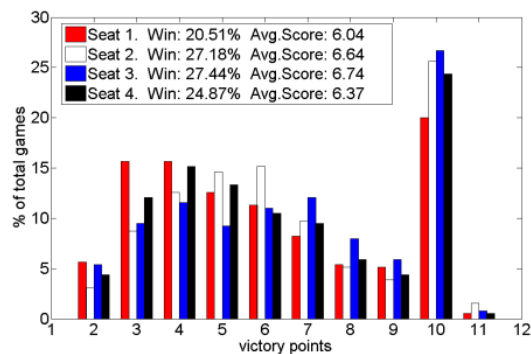


図 5. UCT アクションによる席順の優劣 [15]

発展カードを知る事はできないが、サイコロの出目と同様に相手の持つカードがどのような効果を発動するか非決定的であると考えればよい。

相手の手札を完全には知らないが、相手の情報を完全に知っているものとして近似して考えることにする。これには大きく 2 つの理由がある。

1 つは、サイコロで 7 が出る確率は $1/6$ であり、カードの移動が行われる枚数は 1 枚だけである。これは使用されるカードの頻度と枚数に比べて十分小さい。2 つ目に、やりとりされたカードは多くの場合すぐに使用されるためである。これらの理由から、多くの場合相手のカード情報が完全にわかっている状態であり、わからない状態であってもプレイにほとんど影響することはない。

よって、本研究では他のプレイヤーと交渉を行う上で、重要な要素である相手のカード情報は完全情報であるとする。すなわち、本研究で扱うカタンの AI は交渉を行うときに相手の手札カードを全て知っているものとする。

また、複数の相手と交渉を行なう場合、あるプレイヤーとの交渉条件をもとにして、別のプレイヤーに交渉を提示することが可能となる。これは 2.4 節で述べた通り、相手の戦略を知り、混合戦略を考える必要があるため非常に難しいとされる。本研究では、全てのプレイヤーに対する交渉案の中から、自分の得られる利益の期待値が最大となる交渉を行なう。このとき、相手プレイヤー同士で交渉が行なわれる事はなく、また、自分が交渉できる回数は 1 回限りであるため、交渉するプレイヤーは 1 人だけになる。そのため、プレイを行う環境はマルチエージェントであるが、交渉を行う上ではマルチエージェントシステムとして考えないものとする。

4.4 交渉の方針

本研究において交渉の利益計算方針は 4 つ存在する。また、交渉において提案と承諾について考える必要がある。ただし、交渉は自分のターン 1 回につき 1 回のみ提案するできるものとする。また、交渉案は 1 対 1 交渉について考えるものとする。

受諾プレイヤー

交渉を提案することはない。交渉を受ける場合、相手が提示するすべての交渉案に対して、必ず受諾を選択する。

ランダムプレイヤー

交渉を提案する場合、まず全てのプレイヤーに対して可能な交渉案を列挙する。但し、交渉を行なわないという選択肢もこの交渉案は含んでいるものとする。これら交渉案の中から、均等な確率でランダムに交渉案を提案する。交渉を受ける場合、半々の確率で受諾と拒否を選択する。

ルールベースプレイヤー

交渉を提案する場合、まず全てのプレイヤーに対して可能な交渉案を列挙する。但し、交渉を行わないという選択肢もこの交渉案は含んでいるものとする。これらの交渉案の中から、一定のルールに基づき交渉前と交渉後の手札をポイント付けし交渉案の利益計算を行う。次に、提案手法である、「自己中心的交渉」「和交渉」「積交渉」「利益優先交渉」「交渉受諾優先交渉」のいずれかの評価関数のもと、最も良い評価値の交渉案を提案するものとする。本研究で用いるルールは以下のように設定するものとする。

- 道が引ける場合 3 ポイント
- 家が建てられる場合 10 ポイント
- 街が建てられる場合 15 ポイント
- 自分と交渉相手の点差 例) 自分が 5 点と相手が 7 点の場合、-2 ポイント

お互いが win-win な交渉をもちかけることを前提にすると、同ポイントの場合、自分より点数の高いプレイヤーよりも自分よりも点数の低いプレイヤーに交渉を持ちかける方が勝率があがると予想される。交渉を受ける場合、上記のルールに基づき評価関数を用いて得た評価値がプラスになる交渉案であれば必ず受諾し、0、もしくは評価値がマイナスの場合は拒否するものとする。

UCT プレイヤー

交渉を提案する場合、まず全てのプレイヤーに対して、可能な交渉案を列挙する。但し、交渉を行わないという選択肢もこの交渉案は含んでいるものとする。これら各々の交渉案に対して、N 回のプレイアウトを行い各プレイヤーの勝率を計算し、これをもとにして利益計算を行なう。次に、提案手法である、「自己中心的交渉」「和交渉」「積交渉」「利益優先交渉」「交渉受諾優先交渉」のいずれかの評価関数のもと、最も良い評価値の交渉案を提案するものとする。交渉を受ける場合、交渉を受ける前と受けた後において N 回プレイアウトを行い、勝率を計算する。この勝率をもとに、評価関数を用いて得た評価値がプラスになる交渉案であれば必ず受諾し、0、もしくは評価値がマイナスの場合は拒否するものとする。

5 評価

5.1 交渉を行わない場合の比較

交渉を行なう前に、カタンの開拓者を知的にプレイしない限りは交渉を行なうことすら不可能である。本研究の実験環境であるカタンの開拓者をプレイする実験を行った。まず、全く UCT アルゴリズムによる探索を行わないプレイヤー同士での実験結果を図 6 に示した。このプレイヤーは UCT アルゴリズムを用いていないため、自分のターンで考えられる行動のリストからランダムに選択をおこない行動する（以下、交渉なしランダムプレイヤー）。下記実験では 1,000 回の対戦実験を行いその結果を示す。本実験環境では、1 ゲームあたり 23 秒程度で終わることができる。

結果より、関連研究であった通り [15] 席順による優劣があることが確認された。以降、プレイヤーのアルゴリズムを変更した場合は全ての席順において対戦実験を行い、その平均値の勝率を考えるものとする。次に UCT アルゴリズムをカタンの開拓者に適用した結果を図 7 に示す。UCT アルゴリズムを用いたプレイヤー（以下、交渉なし UCT プレイヤー）は次の自分の行動を UCT アルゴリズムを用いて選択し行動する。交渉を行わないため、交渉に関する行動はこの選択肢には入らないものとする。下記の図の実験では「3 人の交渉なしランダムプレイヤーと 1 人の交渉なし UCT プレイヤー」を用いて実験を行なった。UCT アルゴリズムのプレイアウトの回数を変更する実験を行なった。プレイアウトの回数が多いプレイヤーの方が、プレイアウトの回数が少ないプレイヤーに比べて、より良い行動を選択できると予想される。よって、プレイアウト回数の多いプレイヤーの方がランダムプレイヤーに対して、より良い勝率をおさめる事ができると予想される。下記実験では、席順によって勝率が変化する事から、各々の席順に対して交渉なし UCT プレイヤーを配置し、

「1 人の交渉なし UCT プレイヤーと 3 人のランダムプレイヤー」として 1,000 回の対戦実験を行った。その勝率を示す。

表より、プレイアウトの回数が順に、1 回の場合は UCT プレイヤーに対して 26.42%、100 回の場合は UCT プレイヤーに対して 54.2%、1,000 回の場合は UCT プレイヤーに対して 95.1% の勝率を得た。勝率は各席順の平均値を用いるものとする。4 人プレイを行なっているため、勝率が 25% 以上であればよい。プレイアウトの回数が 1 回の場合はランダムに行動を選択する場合とかわらないため、有意な実力差は得られなかった。しかしながら、プレイアウトの回数を 100 回、1,000 回にした場合、交渉なしランダムプレイヤーは 54.2%、95.1% と優位な結果を得る事ができた。このことから、カタンにおける UCT アルゴリズムは有効であることがわかる。よって、交渉においても、UCT アルゴリズムが有効であると期待される。また、本研究で交渉を扱う場合、交渉なし UCT アルゴリズムでランダムプレイヤーに対して最も良い結果を残した、1,000 回のプレイアウト回数をを用いたカタン AI を用いて、実験を行なうものとする。

5.2 交渉の利益計算方法の比較

交渉の利益計算を行なう方法として受諾プレイヤー、ランダムプレイヤー、ルールベースプレイヤー、UCT プレイヤーを用意した。下記の図 8 はその対戦実験の結果である。現実世界の交渉において、違う種類の資源を使用し、異なる分量で提示する交渉が存在する。カタンにおける交渉でも異なる分量で行なわれる交渉が存在する。例えば、自分は 2 枚資源を提示し、相手は資源を 3 枚提示するといった多対多の交渉が許されている。交渉内容はカード枚数（分量）による影響を受ける可能性がある。1 対 1 交渉の利益計算で最もよい成果を出したルールベースプレイヤーを用いて「3 人の受諾プレイヤーと 1 人のルールベースプレイヤー」の実験を行なった。枚数による影響を考えるため、最も単純な 2 対 1 交渉を考えるものとする。提示する側のプレイヤーは必ず同種または異種のカードが合計 2 枚になるように提示し、受ける側のプレイヤーは 1 種提案するものとする。ルールベースプレイヤーは自分の利益が最大となる交渉案を提示するものとする。ルールベースプレイヤーが自分にとって有利な提案ができている場合は、25% 以上の勝率が期待される。また実験結果より、席順が影響することから全ての席順において 1,000 回対戦実験を行い、得られた勝率の平均値を最終的な勝率

手番／勝数	勝数
1 番手	215
2 番手	315
3 番手	226
4 番手	244
合計	1000
平均	250
標準偏差	38.92942332

図 6. 席順と勝数の比較

手番／プレイアウト回数	1回	100回	1000回
1 番手	10.8%	41.5%	92.9%
2 番手	38.7%	65.8%	98.6%
3 番手	33.9%	55.7%	94.6%
4 番手	22.2%	54.1%	94.4%
平均	26.4%	54.2%	95.1%

図 7. プレイアウト回数による勝率の変化

にしている。結果をの図 8 に示す。

図より、1 対 1 の交渉では勝率が 42.08% だったのに対し、2 対 1 交渉では勝率が 36.55% といずれの場合も有利な交渉を提案できている。しかし、枚数を増やすことによって、勝率が下がっていることが分かる。よって、まずは UCT プレイヤーとルールベースプレイヤーが有利な提案ができているかを検証するために、1 対 1 の交渉において実験するものとする。本研究では「3 人の受諾プレイヤーと 1 人のルールベースプレイヤー」と「3 人の受諾プレイヤーと 1 人の UCT プレイヤー」との対戦実験の行なった。前述の通り、交渉を行なう枚数は 1 対 1 交換のみにしている。UCT プレイヤーの各交渉案に対するプレイアウト回数は 100 回とした。また、ルールベースプレイヤー、UCT プレイヤー共に自分の利益が最大となる交渉案を提示するものとする。また同様に、「3 人のランダムプレイヤーと 1 人のルールベースプレイヤー」と「3 人のランダムプレイヤーと 1 人の UCT プレイヤー」としてランダムプレイヤーとの対戦実験も行なった。ランダムプレイヤーは有利な交渉を提示する上でのベースラインとして考えるものとする。結果より、ルールベースプレイヤーが全受諾プレイヤーに対して 42.08%（前述と同じ）と有意な実力差を残していることから、自分にとって有利となる利益計算を行い、交渉案を提示していることが分かる。一方で、UCT プレイヤーは受諾プレイヤーに対して 23.55% と 25% より低い勝率となった。すなわち、自分にとって有利な利益計算を行なうことができず、結果として有利な交渉案を提示できなかったと考えられる。また、ランダムプレイヤーに対してもルールベースプレイヤーは 41.25% と高い勝率をおさめたが、UCT プレイヤーはランダムプレイヤーに対しても 24.95% と有意な実力差を得る事はできなかった。UCT プレイヤーが良い結果を残せなかった原因を考えていく。ルールベースプレイヤーは UCT プレイヤーに比べて、ランダムプレイヤーと受諾プレイヤーに対し、良い勝率をおさめていることから、ルールベースプレイヤーは UCT プレイヤーよりも良い交渉案を提示していると考えられる。よって、ルールベースプレイヤーの交渉案は正しいと仮定のもと、考えていくものとする。また、UCT アルゴリズムはプレイアウトの回数によって精度を向上できることから、プレイアウトの回数を各交渉案において、100 回の場合と 1,000 回の場合を比べるものとする。UCT プレイヤーとルールベースプレイヤーとの交渉案の一致率を以下に図 9,10 に示す。それぞれ、338,758 回、105,868 回の交渉を行なった。

ルールベースプレイヤーと UCT プレイヤーの交渉案の全体の一致率（交渉を行なわない場合の一致率と、交渉を行なう場合の一致率を合わせた割合）を考える。プレイアウトの回数が 100 回の場合の交渉案の一致率は、ルールベースプレイヤーと UCT プレイヤー共に交渉を行ない同じ交渉案を選択する場合（5.2%）と交渉を行なわない場合（8.2%）を合わせた 13.4% であった。また、プレイアウトの回数が 1,000 回の場合の交渉案の一致率は、ルールベースプレイヤーと UCT プレイヤー共に交渉を行ない同じ交渉案を選択する場合（9.9%）と交渉を行なわない場合（5.0%）を合わせた 14.4% であった。ランダムプレイヤーとルールベースプレイヤーの交渉案の一致率は、6.1% なので、ランダムよりは良い精度で交渉案を選択できているものの、十分でないと考えられる。しかしながら、ルールベースプレイヤーと UCT プレイヤー共に交渉を行なう場合において、プレイアウトの回数を 100 回の UCT プレイヤーはルールベースプレイヤーと 10.2% の一致率であるのに対して、プレイアウトの回数が 1,000 回の UCT プレイヤーは 18.1% と大きく向上した。このことから、プレイアウトの回数を増やすことで、良い交渉案を選択できると考えられる。また、本研究で受諾プレ

	受諾プレイヤー	ランダムプレイヤー
ルールベースプレイヤー (1枚)	42.08%	41.25%
ルールベースプレイヤー (2枚)	36.55%	-
UCTプレイヤー(N=100)	23.55%	24.95%

図 8. 各プレイヤー同士の対戦実験結果

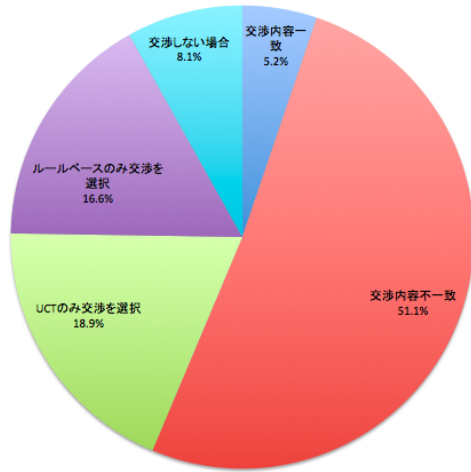


図 9. UCT(N=100) とルールベースの交渉案選択の一致率

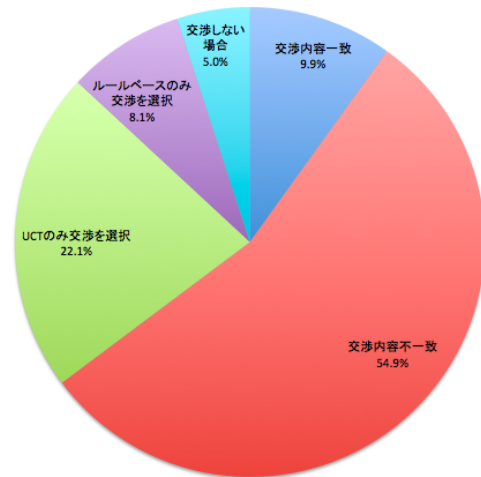


図 10. UCT(N=1000) とルールベースの交渉案選択の一致率

プレイヤーに対して良い結果をおさめられなかった原因として、プレイアウトの回数が十分でなかったと考えられる。しかしながら、100 回のプレイアウトで 1 ゲーム当たり 700 秒程度必要であり、1,000 回プレイアウトを行なった場合、1 ゲーム当たり 6,600 秒程度の時間が必要となる。プレイアウトの回数を増やす事によって線形にプレイ時間が増えてしまうことは大きな課題である。一部の有力でない交渉案をルールベース等で除いてから、探索を行なうといった工夫が必要となる。

また、UCT プレイヤーは序盤では良い交渉案を提案するものの、終盤になるとよい交渉案を選べなくなることがあった。図 11 は UCT プレイヤーとルールベースプレイヤーの交渉案の一致率を縦軸：一致率、横軸：勝率として表したものである。横軸の勝率とは、UCT アルゴリズムでシミュレーションを行なった結果、選択された交渉案において、最も勝率が高かったプレイヤーの勝率を示している。ただし、自分が最も勝率が高かった場合は考えないものとする。

図より、前述の通りプレイアウト回数が 1,000 回の方が 100 回よりも高い一致率となっている。また、いず

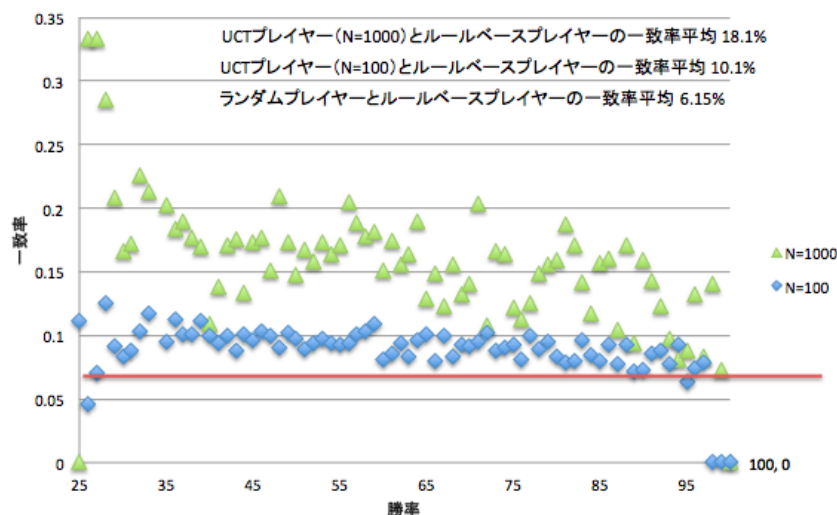


図 11. UCT プレイヤーとルールベースプレイヤーの交渉選択の一致率 (N=100,N=1000)

れもルールベースよりも高い一致率になっていることがわかる。また図より、勝率が高くなるほど、一致率が下がっていく傾向にある。プレイアウト回数が1,000回の場合では、相手プレイヤーの勝率が35%以下の時に平均よりも高い一致率を示しているにも関わらず、勝率が90%以上になると、プレイアウト回数が100回の一致率やランダムプレイヤーに近い一致率になってしまう。これは勝率が拮抗している序盤では良い交渉案を提示しているものの、勝率に差がついてくる終盤になると良い交渉案を選べない原因だと考えられる。相手の勝率が高くなってしまえば、良い交渉を提案できなくなる理由として、自分の勝率計算の精度が悪くなることが考えられる。全ての交渉案に対して1,000回のプレイアウトを行なうが、いずれの交渉案においても一度の交渉で大きく勝率が変わる事はないため、相手の勝率は90%付近となっている。そのため、残りの10%である100回のプレイアウト回数の中から良い交渉案を見つける必要があるが、プレイアウト回数を増やしたにも関わらず、良い一致率を得ることができなくなってしまう。よって、勝率計算において相手の勝率が高くなった場合には一定の重みをつけて計算することで、精度の悪化を防ぐ事が考えられ、今後の課題である。

以上より、UCTプレイヤーの勝率が受諾プレイヤー、ランダムプレイヤーに対して良い結果を残せなかった原因として、「プレイアウト回数が不十分であったこと」「相手の勝率が高くなった場合での精度悪化」が考えられるため、これらの改善を行なうことが課題となる。また、UCTプレイヤーよりもルールベースプレイヤーの方が良い結果を残したことから、ルールベースプレイヤーのほうが良い利益計算を行なっていると考え、交渉の評価選択手法を提案する上で、ルールベースプレイヤーを用いる事にする。

5.3 交渉の評価基準の比較

本研究の提案手法として、良い交渉案を選ぶ評価関数に、以下の5つを示した。自分の利益のみしか考えない「自己中心的交渉」を行なうプレイヤーをベースラインとして扱い、次に、交渉の成功率よりも相手の利益を優先にした「利益優先交渉」と交渉の成功率を優先した「受諾優先交渉」がある。また、お互いの利益をバランスよく配分することで自分の利益と交渉の成功率をバランスよく配分する手法として「和交渉」「積交渉」を提示した。交渉の評価基準を比較する上で、交渉の方針の中で最も良い利益計算を行なっているルールベースプレイヤーを用いることにする。まず、提案を行なうときに、同じ利益計算関数のもと、異なる評価関数を用いて評価実験を行なった。「1人の利益優先交渉プレイヤーと3人の自己中心的交渉プレイヤー」、「1人の受諾優先交渉プレイヤーと3人の自己中心的交渉プレイヤー」、「1人の和交渉プレイヤーと3人の自己中心的交渉プレイヤー」、「1人の積交渉プレイヤーと3人の自己中心的交渉プレイヤー」について対戦実験を行なった。ただし、全てのプレイヤーにおいて受諾するときの評価関数は同一のものとし、その評価関数は「自分の利益が0よりも高くなる交渉は全て受諾する」としている。実験結果を図12に示す。

受諾時に同じ利益関数を使っており、少しでも利益がある場合は受諾するため、自分の評価関数のもとで相手がプラスになる交渉案を提示する、利益優先交渉、受諾優先交渉、積交渉では必ず100%の受諾率となった。和交渉で100%とならなかったのは、相手の利益がマイナスではあるが自分の利益と相手の利益の合計で最も高くなる交渉案を提示する場合があったからだと考えられる。必ず相手が受諾するため、自分にとって最も利益の高い交渉案を常に提案できる利益優先交渉が32.4%と最も高い勝率となった。しかし、受諾優先交渉では自分にとって小さな利益しか得られないにも関わらず相手に大きな利益を与えてしまうため、自己中心的プレイヤーのもとで25%以上の勝率とはならなかった。また、バランス型の評価関数である積交渉が自己中心

	利益優先交渉	受諾優先交渉	和交渉	積交渉
勝率	32.4%	23.7%	30.4%	29.9%
提案成功率	100%	100%	82.3%	100%
受諾率	10.9%	10.9%	11.6%	10.9%

図12. ルールベースの評価値を用いた対戦実験結果

的交渉プレイヤーに対して 29.88% と高い勝率をおさめたことから、自分の利益を一定数確保した交渉案を提示する必要があると考えられる。実際の交渉では、交渉に楽観的で自分の評価では相手にとって利益がない時でも相手プレイヤーが交渉を受諾する場合があります、また一方で、交渉に関して悲観的なプレイヤーでは、相手にとって利益があると自分が考えていた場合でも、受諾しない場合がある。そこで、これらの評価関数が、交渉に対して楽観的な環境と悲観的な環境でも同様の成果が得られるかどうかの評価実験を行なった。交渉を提案するときには、4つの評価手法を用いてそれぞれ3人の自己中心的プレイヤーとの対戦実験を行なった。全てのプレイヤーにおいて受諾するときの評価関数は同一のものとし、その評価関数は楽観的な環境下では「交渉をしない場合の評価値に比べて-10%の交渉案も受諾してしまう」ものとし、また一方で、交渉に関して悲観的なプレイヤーのもとでは、「交渉をしない場合の評価値に比べて+10%の交渉案ならば受諾しない」ものとした。結果を以下の図13、14に示す。

交渉に関して楽観的な環境においては、和交渉のみが30.4%と25%を超える勝率となった。その他の評価関数が自己中心的な評価関数に比べて悪い結果になった原因として、交渉案の受諾に失敗したと考えられる。全てのプレイヤーが楽観的な交渉をおこなった結果、自己中心的なプレイヤーの交渉案を受諾する率が30%程度と楽観的でない場合の11%に比べて大きく上昇した。そのため、自分にとって不利な交渉案を多く受諾してしまう結果となった。和交渉が楽観的な環境下でも高い勝率を残したのは、相手から悪い交渉を行なう中で、自分も相手にとって悪い交渉案を提案していたからだと考えられる。また、交渉に関して悲観的な環境下では、自己中心的交渉に対して全ての評価関数が25%以上の結果を残した。否定的な環境下では、自分にとって不利な交渉案を提示される可能性のある自己中心的な交渉案は受諾されにくいと考えられる。実験結果より、悲観的な環境下では、自己中心的交渉は10%と悲観的でない場合の11%に比べてわずかながら受諾率が下がっていることがわかる。悲観的な環境下では自己中心的な交渉を誤って受諾することがなくなる一方で、自分の交渉案も受諾されにくくなる。そのため、いずれの評価関数でも悲観的でない環境下に比べて相手の受諾率は下がってしまった。利益優先交渉(30.5%)、和交渉(28.3%)では、この影響により悲観的でない場合に比べて勝率が下がってしまったと考えられる。

以上より、交渉を受諾する時、自分では良い交渉案だと思っている交渉案に対して、結果悪い交渉案であった場合があるため、悪い交渉案を受けないよう受諾基準をあげた評価関数が良い結果を残すと考えられる。また、提案する時、実際の交渉では相手の受諾評価基準が自分の評価基準よりもずれていて、楽観的か悲観的かわからないため自己中心的な交渉案を提示するだけでは、高い勝率を残すことはできない。また、評価基準が自分よりもずれていることを利用して、悪い交渉案を混ぜることで勝率を高めることができると考えられる。

これらの実験では受諾するときには、提案するときと同じ評価関数を用いてきた。実際の交渉においては、

	利益優先交渉	受諾優先交渉	和交渉	積交渉
勝率	23.3%	22.6%	30.4%	22.6%
提案成功率	100%	100%	90.2%	100%
受諾率	30.7%	30.5%	30.4%	30.4%

図 13. 交渉に楽観的な環境下での対戦実験結果

	利益優先交渉	受諾優先交渉	和交渉	積交渉
勝率	30.5%	28.4%	28.3%	29.8%
提案成功率	92.5%	93.7%	34.4%	93.6%
受諾率	10.4%	10.4%	10.8%	10.4%

図 14. 交渉に悲観的な環境下での対戦実験結果

	利益優先交渉	受諾優先交渉	和交渉	積交渉
勝率	24.8%	12.8%	28.0%	22.2%
提案成功率	90.4%	93.2%	88.2%	92.1%
受諾率	26.4%	74.0%	44.0%	32.0%

図 15. 各々の評価関数を用いた楽観的な自己中心プレイヤーに対する対戦実験結果

	利益優先交渉	受諾優先交渉	和交渉	積交渉
勝率	22.7%	15.0%	17.7%	20.2%
提案成功率	48.3%	55.9%	37.8%	55.4%
受諾率	26.6%	48.8%	70.9%	32.5%

図 16. 各々の評価関数を用いた悲観的な自己中心プレイヤーに対する対戦実験結果

交渉を行なうプレイヤーは異なる利益計算関数を用いて利益計算を行い、計算された利益を用いて自分の評価関数によって提案、受諾を行なう。お互いの利益計算関数と評価関数が異なっていることにより、相手にとって利益がプラスであるから必ず受諾されると予想し提案した交渉案でも、受諾されないことがある。そのため、同じ評価関数を用いた前述の実験のように提案成功率が 100% になることはないと考えられる。よって、異なる利益計算関数と異なる評価関数を用いた実験を行なう必要がある。本研究では、楽観的な利益計算を行ない、自己中心的な評価関数を用いて交渉案を選択、受諾を行なう「楽観的自己中心交渉プレイヤー」と、悲観的な利益計算を行ない、自己中心的な評価関数を用いて交渉案を選択、受諾を行なう「悲観的自己中心交渉プレイヤー」を用いて、提案手法である 4 つ評価関数の性能評価を行なった。楽観的自己中心交渉プレイヤー、悲観的自己中心交渉プレイヤーは、受諾するとき自分の利益計算関数を用いて得た利益が交渉を行なう前に比べてプラスになる場合は必ず受諾するものとする。また、楽観的な利益計算関数は、交渉を行なう場合の評価値をルールベースの評価値よりも 10% 多く見積もってしまう関数とし、悲観的な利益計算関数は、交渉を行なう場合の評価値をルールベースの評価値よりも 10% 少なく見積もる関数とする。また、「利益優先交渉」「受諾優先交渉」「和交渉」「積交渉」の利益計算関数はルールベースの評価値を用いるものとし、この利益計算関数によって得た利益により、評価関数を用いて提案、受諾を行なう。評価関数によって得た最終的な交渉の期待値がプラスの場合にのみ受諾を行なうものとする。結果を以下の表に示す。

実験結果より、提案成功率と受諾率について考えるものとする。楽観的な交渉プレイヤーに対してはいずれの評価関数も 90% 前後と高い提案成功率を残した。受諾プレイヤーが 12.8% と楽観的環境下の 22.6% に比べて大きく勝率が減少してしまった理由として、提案の受諾率があがったことが考えられる。他の評価関数に比べて、74.0% と高い受諾率になっている。受諾優先交渉では、自分にとって利益が少なくても、相手の利益が大きい交渉案を受け入れてしまうため、受諾率は高かったものの、勝率に結びつくことはなかった。また、悲観的なプレイヤーに対して和交渉では他の実験に比べて 17.7% と低い勝率となった。この結果においても、自分の提案成功率が 37.8% であるのに対し、受諾率が 70.9% と高い受諾率となっている。和交渉の評価関数では、相手の利益と自分の利益の和が交渉を行なう前と後でプラスになっている、かつ、自分にとって利益のある交渉案を受け入れている。そのため、楽観的なプレイヤーに対する受諾プレイヤー同様、相手にとって大きな利益はあるが、自分にとって利益の少ない交渉案を悲観的な交渉プレイヤーに多く提示されたものだと考えられる。大きく受諾率があがったのは、悲観的プレイヤーの利益関数によるものだと考えられる。また、相手に対する提案成功率が 37.8% と楽観的プレイヤーに対して 88.2% だったのに対して大きく減少した。これは、和交渉だけではなく、楽観的プレイヤーに対して 90% 前後の提案成功率を残していた他のプレイヤーにおいても同様のことがいえる。楽観的な交渉プレイヤーに比べて悲観的な交渉プレイヤーでは全体の勝率が

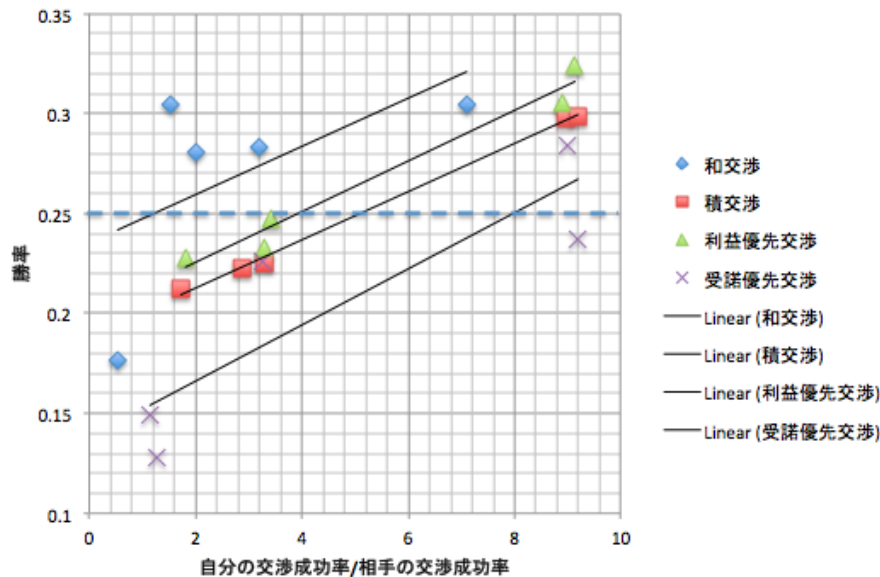


図 17. 評価関数の勝率と交渉成功率

下がったのは、提案成功率が下がったことが原因だと考えられる。以上より、良い交渉を行なう評価関数は、「提案成功率を高める」「受諾率を下げる」評価関数である必要がある。また、敵プレイヤーであった、自己中心的交渉プレイヤーが受諾率は高く提案成功率は低い中で他のプレイヤーに対して高い勝率を残したことから、交渉成功率を考えない純粋な「自分にとっての利益」と「相手に与える損益」の要素が関係していると考えられる。このことから、全ての実験において得られた結果から、交渉の成功率と受諾率の関係の比と勝率の関係を表した図を以下に示す。

図より、全ての評価関数において（自分の交渉成功率/相手の交渉率）の割合が高まれば高まるほど良い勝率を残すことができると考えられる。このことから、交渉における「提案成功率」と「受諾率」が関係していると考えられる。また、和交渉ではばらついたものの、利益優先交渉や、積交渉のように近似直線を描くと一定の直線上にのっていることがわかる。また、図よりこれら 4 つの直線の傾きは一定であるが、評価関数によってその近似直線が上下することがわかる。評価関数によって近似直線が上下する原因として、「自分にとっての利益」と「相手に与える損益」のバランスによるものと考えられる。バランス型の交渉である和交渉では、相手に与える損益を加えたことが高い勝率に結びついたと考えられる。また、受諾優先交渉、利益優先交渉と積交渉では相手に与える損益がないため、自分の得られる利益が大きい順に高い結果を残すことができたと考えられる。利益優先や受諾優先ではなくバランスよく利益配分を行なう和交渉がもっともよい結果を残したことから、「自分にとっての利益」と「相手に与える損益」をバランスよく配分する評価関数が最もよい評価関数だと考えられる。

また、相手プレイヤーによって受諾率や提案成功率が変化することから交渉において相手の評価関数に大きく影響をうけるものと考えられる。相手が楽観的な交渉プレイヤーであれば、相手に与える損益が多い評価関数は、良い結果を残せるが、悲観的な交渉プレイヤーでは悪い結果となってしまう。先行研究 [17] においても、周りの環境によって結果が大きく左右されていた。

以上の実験結果より、交渉において「提案成功率」「受諾率」「自分の利益」「相手の損益」を考えることが必要であることを明らかにした。提案成功率と受諾率は相手プレイヤーの評価関数に依存する。また、「相手の損益」を考慮するのは、自分では悪い交渉案だと思っている交渉案においても、相手との評価基準が自分よりもずれていることを利用して、悪い交渉案を混ぜることで勝率を高めることができると考えられるためである。

6 おわりに

6.1 まとめ

本論文では、交渉の利益計算手法をルールベースと UCT アルゴリズムの 2 つを提案し、UCT アルゴリズムの交渉への適用を試みた。また、交渉の評価関数を 5 つ提案し、対戦実験を行なう事で、交渉における 4 つの要素を明らかにした。UCT アルゴリズムによる利益計算はプレイアウトによる勝率のフィードバックを用いることにより行なった。受諾プレイヤーに対して有意に高い勝率を残したルールベースプレイヤーと交渉案の一致率を確認することにより、UCT アルゴリズムによる利益計算の問題点を明らかにした。また、交渉の評価関数の実験では、利益を優先する評価関数と交渉の成功率を高める評価関数とこれらをバランスよく配分した評価関数を用意し、自己中心的なプレイヤーに対して様々な環境において検証を行なった。交渉においては、自分の利益計算関数と評価関数が相手の利益計算関数と評価関数にずれが生じていることから、自分では悪い交渉案だと思っている交渉案においても、相手との評価基準では、良い交渉案だと判断する場合があるため、悪い交渉案を混ぜることが有効であった。このことから、「提案成功率」「受諾率」「自分の利益」のみならず、「相手の損益」を考慮した評価関数を生成する必要があると考えられる。

6.2 今後の課題

交渉を行なうに上で

- 全てのプレイヤーに対して可能な交渉案のリストアップ
- 各交渉案の利益の見積もり
- 評価関数による各交渉案の期待値計算
- 交渉案の提示

の順番で行なわれる。本研究では可能な交渉案をリストアップするときに、問題を簡単化するために、1 対 1 交渉に限定した。現実世界での交渉では、提示する交渉案は資源の種類のみならず、交渉の量が異なる交渉案が提示される。よって、多対多の交渉モデルを構築する必要がある。また、本研究では各交渉案の利益計算を行なうために UCT アルゴリズムを適用することを試みた。UCT アルゴリズムにおいて良い結果を残せなかった原因として、シミュレーション回数と負けているときに自分にとって有利な交渉案を提示できなかった。シミュレーション回数は指数的に増加してしまうことから、最後までプレイアウトを行わず、一定の時間でシミュレーションを打ち切り、新しく評価関数を用意して盤面を評価し、その時点で評価値が最も高いプレイヤーを暫定の勝利者とするといった工夫が必要となる。また、負けているときにはシミュレーション回数を増やす事で精度を高められなかったことから、重み付けを行なうことによって精度を保つ必要があり、パラメータを調整することで、この重み付けの方法を調査する必要がある。また、本研究では交渉案を提示する回数を 1 回とし、1 回で自分の利益を最大にする評価値を返す評価関数を提案した。しかし、実際の交渉では、何回か交渉を行なう事でお互いの妥協点を探っていく事になる。さらに、妥協点を探すときには、先行研究にあるとおり、悪い交渉案を先に提示しておいて誤った認識を相手に与える事で有利な交渉を行なうといった方法が考えられる。また、時間や交渉回数が限られてくる場合には、交渉を行なう手順が複雑となる。

謝辞

本研究を進めるにあたって、多くの方にお世話になりました。指導教員である近山隆教授には研究の方向性に始まり、様々な面でアドバイスをいただきました。また、研究に対する心構えを教えてくださいました。同じく、指導教員である鶴岡慶雅准教授には、研究の内容や進め方について数多くのアドバイスをいただきました。マンチェスター大学の三輪誠さんには、技術的なアドバイスや論文紹介など細部にわたって助けていただきました。研究室博士の浦晃さんには、本研究で行なうべき実験の方向性に関してアドバイスをいただきました。研究室の皆様には、研究生活のみならず、技術的な面でも大変お世話になりました。多大な支援をいただきありがとうございました。この場を借りて厚くお礼申し上げます。

参考文献

- [1] Roger Fisher and William L Ury. *Getting to yes: Negotiating agreement without giving in*. Penguin, 2011.
- [2] Roy J Lewicki, Bruce Barry, and David M Saunders. *Essentials of negotiation*. Irwin/McGraw-Hill New York, 2001.
- [3] 印南一路. ビジネス交渉と意思決定. 日本経済新聞社, Vol. 45, No. 7, pp. 497–499, 2002.
- [4] 安村禎明, 秋山英久, 小口邦彦, 新田克己. モノポリーゲームにおける交渉エージェント. 情報処理学会論文誌, Vol. 43, No. 10, pp. 3048–3055, 2002.
- [5] 川越敏司. 行動ゲーム理論入門. NTT 出版, 2010.
- [6] Victor Sanchez Anguix. *Complex Negotiations in Multi-Agent Systems*. PhD thesis, Universitat Politècnica de Valencia, Spain, 2013.
- [7] 森口竜平, 日湯淳子, 小山田裕太, 斎藤誠一, 城仁士. 交渉論. *Bulletin of the Graduate School of Human Development and Environment Kobe University*, Vol. 3, No. 1, 2009.
- [8] Levente Kocsis and Csaba Szepesvari. Bandit based Monte-Carlo planning. In *Machine Learning: ECML 2006*, pp. 282–293. Springer, 2006.
- [9] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, Vol. 47, No. 2-3, pp. 235–256, 2002.
- [10] Luca Branca and Stefan J Johansson. Using multi-agent system technologies in Settlers of Catan bots. *City*, Vol. 2, p. 3, 2007.
- [11] Nathan R Sturtevant. An analysis of uct in multi-player games. In *Computers and Games*, pp. 37–49. Springer, 2008.
- [12] John Nash. Non-cooperative games. *The Annals of Mathematics*, Vol. 54, No. 2, pp. 286–295, 1951.
- [13] Frederik Christiaan Schadd. *Monte-Carlo search techniques in the modern board game Thurn and Taxis*. PhD thesis, Maastricht Univ., Netherlands, 2009.
- [14] Markus Guhe and Alex Lascarides. Trading in a multiplayer board game: Towards an analysis of non-cooperative dialogue. In *Proceedings of the 34th Conference of the Cognitive Science Society*, 2012.
- [15] István Szita, Guillaume Chaslot, and Pieter Spronck. Monte-Carlo tree search in Settlers of Catan. In *Advances in Computer Games*, pp. 21–32. Springer, 2010.
- [16] カタンの開拓者 AI の Jettlers のページ. <http://nand.net/jsettlers/devel/>.
- [17] Takayuki Ito, Minjie Zhang, Valentin Robu, and Tokuro Matsuo. *Complex automated negotiations: theories, models, and software competitions*. Springer, 2012.