

過去交渉情報を元にK-近傍法を用いた 自動交渉エージェントの試作

奥原 俊^{1,a)} 濱田 大槻^{1,b)} 伊藤 孝行^{1,c)} アーメッド ムスタファ^{1,d)}

受付日 2019年1月25日, 採録日 2019年7月3日

概要: マルチエージェントシステムの研究分野において自動交渉エージェントが注目を集めている。自動交渉エージェントの研究では、互いに相手の効用情報を知ることができない状況下における二者間交渉問題が重要な課題となっている。二者間交渉問題に対して、本論文では過去の交渉情報に基づく自動交渉エージェントを提案する。提案するエージェントは過去の交渉情報を参照し、K-近傍法による合意案候補探索を行うことにより社会的余剰値を高めるような合意案の提案を行う。また、評価実験として自動交渉の国際競技会である PRIANAC、および ANAC 決勝大会出場エージェントとの交渉シミュレーションを行った。提案するエージェントは様々な交渉ドメインにおいて有効な交渉結果を得られることを示す。

キーワード: マルチエージェントシステム, 自動交渉, 自動交渉競技会

An Implementation of Automated Negotiating Agent based on K-nearest Neighbor Algorithm

SYUN OKUHARA^{1,a)} HAMADA DAIKI^{1,b)} TAKAYUKI ITO^{1,c)} AHMED MOUSTAFA^{1,d)}

Received: January 25, 2019, Accepted: July 3, 2019

Abstract: A automated negotiating agent attracts attentions in the field of multi-agent system. Bilateral multi-issue closed bargaining problems are critical in the field of automated negotiations. In this paper, we propose an agent that makes offers by using the negotiation history. Then agent learns the degree of agreement with this negotiation history data by using the K-nearest Neighbor Algorithm. Our agent succeeded to increase the social welfare score by searching agreement candidate based on the proposed algorithm. We demonstrate that our agent outperforms the existing agents in the past negotiating agent competitions includes the PRIANAC and the ANACs.

Keywords: multi-agent system, automated negotiation, pacific rim international automated negotiation agents competition

1. はじめに

マルチエージェントシステムの研究分野において自動交渉エージェントが重要な研究課題として注目されている。自動交渉エージェントは数値化した選好情報をもとに、人

間の代理として交渉を行うことを目的としている。自動交渉は現実世界においても、異なる効用を持つ人工知能どうしを協調させるために必要な機構であり、将来的には、高度な交渉戦略を持つエージェントの合理的な判断によって、人間には解決困難な複雑で大規模な交渉問題においても合理的な合意形成の支援を行うことができる。自動交渉エージェント技術の応用としては電子商取引システム [11], スケジューリングシステム [12], スマートファクトリの自動化があげられる。

自動交渉エージェント研究の国際的な取り組みとして、自動交渉エージェント国際競技会 (Automated Negotiating

¹ 名古屋工業大学
Nagoya Institute of Technology, Nagoya, Aichi 466-8555, Japan

a) okuhara@itolab.nitech.ac.jp

b) hamada.daiki@itolab.nitech.ac.jp

c) ito.takayuki@nitech.ac.jp

d) ahmed@nitech.ac.jp

Agents Competition : ANAC) [2], [17] が, International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS) などと共催で 2010 年より開催されている. ANAC が対象とする交渉問題は, 互いの選好情報を明かさないという条件のもとで, 時間経過によって得られる効用が減少する割引効用や, 合意に失敗した場合においても一定の効用値を得ることができる留保価格などが設定されており, 現実の交渉に近付ける取り組みがされている. ANAC では多様な自動交渉エージェントが, より良い交渉結果を得るために様々な合意案候補探索手法を用いている. 本論文における良い交渉結果とは, 過去の交渉履歴を明示的に利用して, 早期合意形成を行い, 社会的余剰値 (交渉参加者の獲得効用値の総和) が大きな合意案を得ることである.

既存の研究 (たとえば文献 [6], [8], [13]) でも, 過去の交渉履歴を利用し機械学習を応用した例はない. これは公式競技会である ANAC や PRIANAC において, 過去の交渉履歴の活用が禁止されていた影響もある. 2018 年に過去の交渉履歴が活用可能にルール変更されたため, 本研究では過去の交渉履歴を利用し機械学習を応用する方法を提案している.

ANAC 同様に, 二者間複数論点交渉問題を取り扱っている競技会, PRIANAC (Pacific Rim International Automated Negotiating Agents Competition) 2018 が 2018 年に開催された. PRIANAC では ANAC と同様なルールに加え, 自動交渉エージェントの国際競技会では初めて, 機械学習的なアプローチが推奨されている.

PRIANAC2018 では, 機械学習の効果を確かめるために, 過去の交渉データを使えるようなルールの拡張が行われた. 具体的には, ローカルファイルを利用して交渉のデータを保存することができる. これにより, 同じ交渉相手に対して, 過去の交渉データを使った交渉戦略を構築できる. 一方, 過去の ANAC では毎回の交渉は基本的には独立しており, 1 回の交渉のデータに基づいてしか交渉を行うことはできない.

本論文では, 過去の交渉が利用できるような状況に対して効果的なエージェントを提案する. 提案するエージェントは, 合意案候補の探索手法として K-近傍法を用いた合意案候補の探索を行う. 本論文で提案するエージェントを評価するために, PRIANAC2018, ANAC 決勝大会出場エージェントと交渉シミュレーションを行い, 交渉結果から評価スコアを導出する.

本論文の構成を次に示す. 2 章で自動交渉エージェントの関連研究について取り上げる. 3 章で PRIANAC2018 の概要と交渉ルールについて述べる. 4 章で本論文で提案するエージェントの合意案候補の探索手法について述べる. 5 章で提案するエージェントと PRIANAC2018, および ANAC 決勝大会出場エージェントによって交渉を行い,

評価スコアを比較する. 6 章で本論文のまとめと今後の課題を示す.

2. 関連研究

本章では, 自動交渉エージェントの交渉戦略に関連する研究について述べる.

まず ANAC が開催される以前の研究として Faratin らは, 時間によって譲歩の大きさが変化する譲歩関数を提案した [5]. Faratin らの譲歩関数はアドホックな方法であり, パラメータを設定することによって交渉の経過時間 t による譲歩の割合が変化する. パラメータに依存するため, 交渉相手に関する情報を事前に得ることができない状況下では交渉が難しくなる. また ANAC の競技会において具体的に利用された戦略ではない.

以下, ANAC の競技会に参加したエージェントについてのべる. まず, 過去の交渉履歴を用いることができない場合の代表的なエージェントについてのべる. 1 つの交渉において, その過程で得られる相手の提案履歴に対して適切な譲歩を推定する手法も提案されている. AgentK [9] は ANAC2010 の優勝エージェントである. AgentK の特徴として, 相手の提案履歴から相手の選好情報と交渉姿勢を推測する点があげられる. The Fawkes [1] は ANAC2013 の優勝エージェントである. The Fawkes の特徴として, 相手の提案履歴に対して離散ウェブレット予測 [4] と呼ばれる学習手法により譲歩の度合いを推定する. Atlas3 は ANAC2015 の優勝エージェントである. Atlas3 の特徴として, 交渉戦略を戦略型ゲームとして解析することによって適切な譲歩関数を設計する点があげられる. さらに, オンライン学習 [3] やガウス過程 [16] に基づいた手法も提案されているが, 以上のエージェントは過去の交渉履歴を用いることは行っていない. これは, ANAC の競技会では, 過去の交渉の情報を保存することが許されず, 基本的には現在の交渉における情報だけを使うことが許されていたためである. 一方, PRIANAC2018 の社会的余剰部門で優勝した Agent33 [10] は交渉履歴を用いるエージェントの 1 つである. 交渉相手の提案履歴から提案の偏差を求め交渉相手の選好情報を推定する探索手法をとる.

文献 [6] では, 交渉において相手の効用関数を推定するために, 複数の基礎的な効用関数の組合せとその重みづけを 1 つの交渉セッションの中で行う方法を提案している. 本提案との違いは, 本提案は過去の交渉履歴を利用しているが, 文献 [6] では利用していない点である.

文献 [13] では, 人間のダイアログから深層学習を用いてその効用関数を推定し, ナッシュ解を予測する方法が提案されている. 自動交渉としての交渉フレームワークの定義が異なる点と, 過去の交渉履歴を利用していない点が, 本論文で提案する手法とはまったく異なる.

文献 [8] では, 過去に行った交渉から得られた学習モデ

ルを、将来の交渉に転移学習として適用する方法が提案されている。文献 [8] では、学習自体は 1 つの交渉セッションの中で閉じており、本論文で提案するような過去の交渉履歴を取り入れ、現在の交渉に応用することはしていない。

本論文で提案するエージェントの合意案候補探索手法として K-近傍法を用いた合意案候補の探索を行う。合意案候補の探索を行うにあたり、過去の交渉情報を元に交渉相手から提案された合意案、および合意した合意案を AgreedBid として扱い自身が提案して否定された合意案を RejectedBid として扱い、一定の効用値を獲得できる自身の合意案候補群を 2 クラスに分類し、新しい AgreedBid 群を作成し合意案候補を探索する。分類する際に使用する特徴量として、交渉相手が提案した時間、合意案候補の効用値、以上の 2 つを特徴量して扱う。したがって、提案するエージェントは既存手法よりも安定して早期合意形成を行い社会的余剰値を高めることができる。

特徴量として自身の正規化効用値および正規化された交渉時間を用いている。この意図するところは、ANAC や PRIANAC では、相手に効用空間をいっさい明かさない (Closed Negotiation) という得られる情報が少ない設定をしており、その設定下で得られる情報としては、自身の正規化効用値と正規化された交渉時間の 2 つくらいしか特徴量がないためである。

3. 自動交渉エージェント競技会 (PRIANAC)

3.1 PRIANAC の概要と目的

PRIANAC は交渉を研究する世界中の研究者たちが自動交渉エージェントを作成し、作成されたエージェントによるトーナメントの評価スコアを競い合う国際競技会である。PRIANAC の目的は次のようになる。

- 選好情報が明らかでない相手に対して、様々な状況において合理的に対応できる実用的な自動交渉エージェントの設計
- 多様な交渉戦略の客観的な評価指標の提供
- 交渉過程におけるエージェントの学習や適応戦略、および、未知の交渉相手のモデリング手法の探求
- 最先端の自動交渉エージェントと交渉シナリオの収集と提供

本論文では、提案するエージェントの比較対象として PRIANAC と ANAC 決勝大会に出場したエージェントを使用する。

3.2 交渉ルール

【交渉ドメイン】

交渉ドメインとは現実世界の交渉問題をコンピュータプログラムであるエージェントが扱うことができる数値として定義したものである。二者間多論点クロード交渉問題 (BMCBP: Bilateral Multi-issue Closed Bargaining

Problems) における交渉ドメインでは交渉問題を論点ごとに分解し、各論点のとりうる選択肢を数値に変換している。PRIANAC では、交渉に参加するすべてのエージェントに交渉ドメインの完全な情報が与えられる。交渉ドメインにおける論点の集合を $\mathcal{I}$ とする。個々の論点 i ($i \in \mathcal{I}$) はそれぞれ $[0, v_i]$ の範囲の整数値をとる (v_i は論点 i のとりうる最大値)。合意案候補集合を $\mathcal{S}$ 、個々の合意案候補を $\vec{s}$ ($\vec{s} \in \mathcal{S}$)、そして、論点数が n である場合、 $\mathcal{S}$ は式 (1) で定義される。

$$\mathcal{S} = \{\vec{s} = (s_1, s_2, \dots, s_n) \in \mathbb{N}^n \mid 0 \leq s_i \leq v_i, i = 1, 2, \dots, n\} \quad (1)$$

【効用関数】

効用関数とは合意案によって交渉者が得ることができる効用値を定義した関数である。本論文では Ito らによって提案された制約に基づく効用関数 [14] を用いる。制約に基づいた効用関数について説明する。交渉問題において、交渉者は個々にユニークな制約集合を持つ。合意案候補 $\vec{s}$ によって充足される制約集合を $\mathcal{C}(\vec{s})$ 、個々の制約を c_k ($c_k \in \mathcal{C}(\vec{s})$) と表す。制約 c_k は単一、もしくは複数の論点に関して制約充足条件となる値の範囲、および効用値を持つ。合意案候補 $\vec{s}$ は制約 c_k を充足するため、交渉者は評価関数 $w(c_k, \vec{s})$ によって効用値を得ることができる。制約 c_k に対する標準化係数を β_k としたとき、 $[0, 1]$ で正規化された効用関数 $U(\vec{s})$ は式 (2) で定義される。

$$U(\vec{s}) = \sum_{c_k \in \mathcal{C}(\vec{s})} \beta_k \cdot w(c_k, \vec{s}) \quad (2)$$

PRIANAC2018 や ANAC 競技会では、効用は $[0, 1]$ に正規化されており、本論文では正規化効用と呼ぶ。

【交渉プロトコル】

本論文では Rubinstein が提唱する Alternating Offers [15] をベースとした交渉プロトコルを用いる。Alternating Offers に関する研究は多く存在し、本論文とは前提や交渉条件が異なるが、ゲーム理論 [14] や機械学習に基づくアプローチが提案されている [3]。Alternating Offers では次の 3 つのアクションによって合意案候補 (Bid) を扱う。図 1 は Alternating Offers におけるエージェントと 3 つのアクションの関係を示した概要図である。

- Offer: 相手に Bid を提案するアクション。最初に Offer を行うことができる交渉者は交渉開始前に設定する。
- Accept: 相手が Offer してきた Bid を受容するアクション。Accept した場合は交渉が成立し、互いに合意案を自身の効用関数で評価し、評価した値に割引効用を適用した効用値を獲得して交渉を終了する。相手が Offer した Bid を拒否する場合は、自身が新たな Bid を代替案として相手に Offer する。

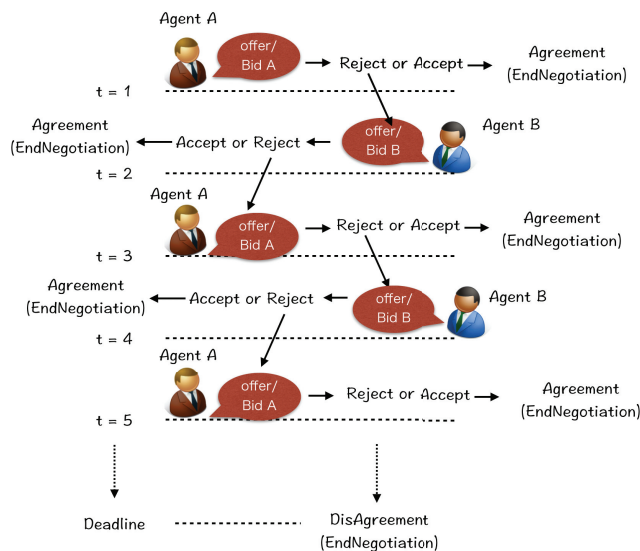


図 1 Alternating Offers の概要図

Fig. 1 Overview of Alternating Offers.

- *EndNegotiation*: 交渉を放棄するアクション。二者間交渉の場合はいずれかの交渉者が *EndNegotiation* を選択した時点で交渉は終了する。 *EndNegotiation* が選択された場合、交渉者は留保価格に割引効用を適用した効用値を獲得し、交渉を終了する。

Alternating Offers における交渉の例として、エージェント A とエージェント B による交渉を考える。最初の行動権をエージェント A が持つ場合、まずエージェント A が任意の合意案候補である Bid_A をエージェント B に Offer する。エージェント A の Offer の後、行動権がエージェント B に移る。エージェント B はエージェント A が Offer した Bid_A を受容するならば *Accept* する。エージェント B は、 Bid_A を拒否するならば代替案として任意の合意案候補である Bid_B をエージェント A に Offer する。エージェント B の Offer の後、再び行動権がエージェント A に移る。本例のような手続きを繰り返すことによって二者間交渉は進行する。また、行動権を持つ交渉者は任意で *EndNegotiation* を選択することができる。

【制限時間】

現実世界の交渉問題は有限の交渉時間内に結論を導くことが要求される。したがって、本論文では制限時間が存在する交渉問題を想定する。なお、制限時間を超過した場合、交渉者は *EndNegotiation* を選択する。本論文では交渉開始からの経過時間を $[0, 1]$ の正規化時刻 t として扱う。

【割引効用】

本論文では、交渉条件として時間経過によって得られる効用値が減少する割引効用を考慮する。割引効用は既存研究 [15] でも考慮されている交渉条件である。本論文における割引効用は、効用関数によって得ることができる効用値に対して $(0, 1]$ の値をとる割引係数に基づいて算出した値となる。割引係数 df は交渉問題ごとに設定されている。

合意案候補 $\vec{s}$ の効用値を $U(\vec{s})$ とした場合、割引効用を考慮した効用関数 $U_D(\vec{s}, t)$ は式 (3) で定義される。

$$U_D(\vec{s}, t) = U(\vec{s}) \cdot df^t \quad (3)$$

【留保価格】

本論文では、*EndNegotiation* を選択した場合に一定の効用値を得ることができる留保価格が存在する。留保価格は交渉問題ごとに設定されている。留保価格にも割引効用は適用され、留保価格 RV に割引効用を適用した留保価格 $RV_D(t)$ は式 (4) で定義される。

$$RV_D(t) = RV \cdot df^t \quad (4)$$

4. K-近傍法に基づく合意案候補探索

4.1 合意案候補の提案

本論文で提案するエージェントは、社会的余剰値を高め早期合意形成を目的として合意案候補の提案を行う。次の手順にしたがって合意案候補を交渉相手に提案 (図 1 における Offer) する。

手順 1 交渉を行う際に交渉相手から提案された、また、その交渉において実際に合意できた合意案を *AgreedBid*、自身が提案して否定された合意案を *RejectedBid* として交渉履歴に保存する。

手順 2 一定以上の効用値を獲得できる合意案候補群を取得し、K-近傍法を使用して取得した合意案候補群を *NewAgreedBid* 群と *NewRejectedBid* 群に分類する。手順 2 に関しては 4.3 節で詳細を説明する。

手順 3 分類して得られた *AgreedBid* 群の中から交渉相手の提案履歴の頻度、および近傍に提案された合意案を元にパレート改善を行い、最終的な合意案候補群を獲得する。

4.2 交渉経過時間を考慮した合意案候補の探索

PRIANAC では 1 回の交渉を 10 秒で行い同じ交渉条件で交渉を 100 回繰り返すため、参照できる情報量が多いので交渉相手の選好情報を特定しやすい。参照できる情報は以下のものである。

- 交渉参加者の提案応答履歴
- 合意結果における経過時間
- 交渉における合意案

PRIANAC では 1 回の交渉が 10 秒という時間で行われるため、効率的に合意案候補の探索を行う必要がある。既存エージェントの合意案候補の探索手法は焼きなまし法 (SA) による合意案候補探索と近傍探索、頻度探索によるパレート改善を行うことで合意案候補は探索される。既存手法の場合、現在実行している交渉の交渉情報のみを参照しパレート改善を行うため、交渉終盤にならなければ十分な交渉情報を得ることができない。よって、既存手法では早

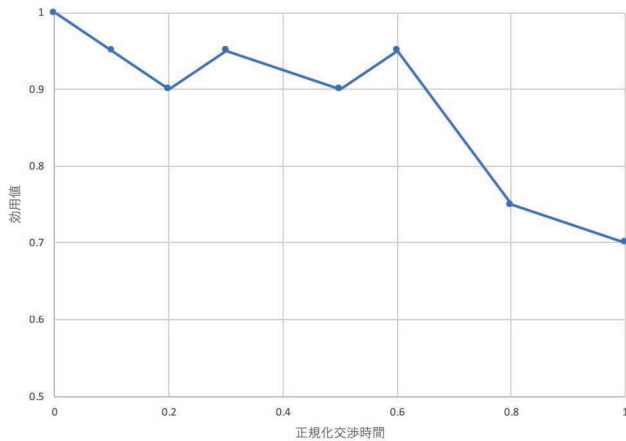


図 2 交渉時間に基づく合意案候補の効用値の推移

Fig. 2 Transition of utility value of candidate agreement proposal based on negotiation time.

期合意形成を行うことができなくなる。またエージェントは自身の閾値以上の合意案候補を獲得し交渉相手に *Offer* するため、交渉経過時間により合意案候補の優先度も変化する。そのため、全体の効用を高めつつ早期合意形成を行うためには、交渉経過時間も考慮した合意案候補探索が必要となる。図 2 はエージェントの提案した合意案の効用値の推移である。図 2 が示すようにエージェントは交渉経過時間によって、提案する合意案が変化する。

4.3 初期の合意案候補の作成

本エージェントは、以下の (1) と (2) の 2 つの方法で初期の合意案候補の集合を生成する。

(1) 過去の入札履歴のなかの入札の中から、あらかじめ設定した最大閾値と最小閾値の間の効用を得られ、かつ経過時間による閾値を上回る効用を得られる入札の集合を初期の合意案候補の集合とする。

- 最大閾値と最小閾値は以下の 9 つのペアからランダムに選ぶ。9 つのペアは経験的に与えた値である。(最大, 最小) = (0.90, 0.85), (0.85, 0.825), (0.825, 0.80), (0.80, 0.775), (0.775, 0.750), (0.750, 0.725), (0.725, 0.70), (0.70, 0.675), (0.675, 0.650)
- 経過時間による閾値 (TimeTh) は以下の式で与えられる。

$$TimeTh = \begin{cases} 0.7 - 0.1 \times time & (time \geq 0.5) \\ 0.7 + 0.4 \times time & (otherwise) \end{cases}$$

(2) (1) の方法で得られる入札がなかった場合は、ランダムに 1 つ合意案候補を生成する。

4.4 K-近傍法による合意案候補群の分類

提案手法では交渉参加者の提案応答一覧を獲得し、交渉相手が *Offer* した合意案と本論文で提案するエージェントが *Offer* した合意案を以下の 2 つの合意案としてラベル付

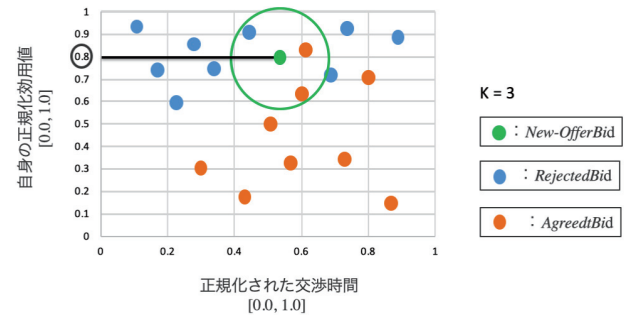


図 3 K-近傍法による合意案候補の分類の概要図

Fig. 3 Overview of classification of agreement proposal candidates by K-nearest neighbor algorithm.

けを行い保存する。

AgreedBid : 交渉相手が *Offer* した合意案、または実際にある交渉において合意した合意案

RejectedBid : 本論文で提案するエージェントが *Offer* して合意することができなかった合意案

本論文で提案するエージェントは、K-近傍法による合意案候補の探索を行う。手順 1 で先ほど説明したように交渉相手との交渉で得た *AgreedBid*, *RejectedBid* の記録を保存する。そして手順 2 で一定以上の効用値を獲得できる合意案候補群を *AgreedBid*, *RejectedBid* に分類する。K-近傍法による分類を行うので、提案応答として保存されている合意案 (*AgreedBid*, *RejectedBid*) と合意案候補群のすべて合意案のユークリッド距離を算出する必要がある。分類する際に使用する特徴量として以下の 2 つの要素を使用する。

Time : 交渉者がある合意案を *Offer* した時間

Utility : 提案エージェントが *Offer* された合意案、また自身が *Offer* した合意案を提案エージェントの効用空間において効用値を算出して獲得できる効用値

図 3 に K-近傍法による合意案候補の分類の概要図を示す。

一定以上の効用値を獲得できる合意案の総数を n 個、合意案候補群に含まれる合意案を B_i ($i = 1, 2, \dots, n$), 合意案が *Offer* された時間を B_i^{Time} ($i = 1, 2, \dots, n$), 合意案の効用値を B_i^{Util} ($i = 1, 2, \dots, n$) とし、また、過去の交渉の提案応答一覧として保存されている合意案の総数を m 個、保存されている合意案を RB_j ($j = 1, 2, \dots, m$), 合意案が *Offer* された時間を RB_j^{Time} ($j = 1, 2, \dots, m$), 合意案の効用値を RB_j^{Util} ($j = 1, 2, \dots, m$) としたとき、ある合意案 B_i に対してユークリッド距離 d_i ($i = 1, 2, \dots, n$) は式 (5) で導出される。

$$d_{ij} = \sqrt{(B_i^{Time} - RB_j^{Time})^2 + (B_i^{Util} - RB_j^{Util})^2} \quad (5)$$

式 (5) で導出されたユークリッド距離をソートし、一定以上の効用値を獲得できる合意案候補群を新しい *AgreedBid* 群, *RejectedBid* 群に分類する。新たに得られた *AgreedBid*

群中から再び合意案候補の探索を行いパレート改善を行う。

K-近傍法による合意案候補の探索は次のような処理で行われる。アルゴリズムの詳細を Algorithm 1 に示す。

- (1) 過去の提案応答一覧から合意案候補群を獲得し (1 行目), 一定以上の効用値を獲得できる合意案候補群を獲得する (2 行目)。
- (2) 手順 2 で獲得した合意案候補群の合意案と手順 1 で獲得した合意案候補群の合意案とのユークリッド距離をすべて算出する (7 行目から 13 行目)。
- (3) ユークリッド距離を算出し, 新たに AgreedBid と分類された Bid を NewAgreedBidlist に追加する (14 行目から 23 行目)。
- (4) NewAgreedBidlist の合意案をパレート改善するための関数を呼び出す (24 行目)。

Algorithm 1 K-近傍法による合意案候補の探索アルゴリズム詳細

```

1: PastBidlist: 過去の提案応答一覧として保存されている合意案候補群 (GENIOS から提供される)
2: Bidlist: 前もって定義した以上の効用値が獲得できる合意案候補群
3: NewAgreedBidList: 新たに AgreedBid として分類された合意案候補群
4: Function KnnBidSearch:
5:   NewAgreedBidList := {}
6:   dis_list := {}
7:   for  $i = 0$  to Bidlist.size do
8:     for  $j = 0$  to PastBidlist.size do
9:        $Bid = Bidlist[i]$ 
10:       $PastBid = PastBidlist[j]$ 
11:       $dis\_list.add(EuclidDistance(Bid, PastBid))$ 
12:    end for
13:  end for
14:   $sort(dis\_list)$ 
15:   $k := \sqrt{dis\_list.size}$ 
16:  for  $m = 0$  to  $k$  do
17:    if  $dis\_list[m].includedinAgreeBid$  then
18:       $agreedcount++$ 
19:    end if
20:  end for
21:  if  $agreedcount > k/2$  then
22:     $NewAgreedBidList.add(Bid)$ 
23:  end if
24: パレート改善のため getBidbyNeighborhoodSearch と getBidbyFrequencySearch を呼ぶ。
25:
```

4.5 合意案候補のパレート改善

交渉において交渉相手の合意を得るためには, 自身のみでなく, 相手にとっても効用値が大きな合意案候補を提案する必要がある。既存エージェントの場合は自身の効用関数のみ考慮した合意案候補の探索を行った後に近傍探索と頻度探索によるパレート改善を行うが, 提案エージェントはあらかじめ K-近傍法により相手にも有利となるような合

意案候補群を交渉時間に基づいて作成した後にパレート改善を行うので, 社会的余剰値を高めつつ早期合意形成が可能となる。提案エージェントは手順 3 ではパレート改善手法として近傍探索と頻度探索によるパレート改善を行う。

【近傍探索によるパレート改善】

パレート改善ではまず近傍探索を行う。近傍探索とは, 探索範囲を基準となる合意案候補の近傍に限定する局所探索である。近傍探索は相手が提案した合意案候補が充足する制約を, 相手が提案した合意案候補の近傍もまた充足するというヒューリスティックスに基づいている。近傍探索による合意案候補の探索の処理の概要とアルゴリズム詳細を以下に示す。

- (1) K-近傍法で分類した新たな合意案候補群を NewAgreedBidlist とする (1 行目)。
- (2) 基準となる合意案候補を, 最後に相手が提案した合意案候補 *mOfferedBid* とする (2 行目)。
- (3) *mOfferedBid* と NewAgreedBidList を比較し, 選択肢が同じであれば *matchingcount* をインクリメントする (7 行目から 19 行目)。
- (4) *matchingcount* が論点数の半数以上であれば *NeighborhoodBidlist* に追加する (16 行目から 18 行目)。

Algorithm 2 近傍探索による合意案候補の探索アルゴリズム詳細

```

1: NewAgreedBidList: 新たに AgreedBid として分類された合意案候補群
2: mOfferedBid: 相手から Offer された合意案
3: IssueList: 論点のリスト
4: NeighborhoodBidList: 相手から Offer された合意案と類似度の高い合意案候補群
5: Function getBidbyNeighborhoodSearch (NewAgreedBidList):
6:   NeighborhoodBidList := {}
7:   for  $i = 0$  to NewAgreedBidList.size do
8:      $AnAgreedBid = NewAgreedBidlist[i]$ 
9:     for  $j = 0$  to issuelist.size do
10:       $offeredValue = mOfferedBid.getValue()$ 
11:       $agreedValue = AnAgreedBid.getValue()$ 
12:      if  $offeredValue == agreedValue$  then
13:         $matching\_count++$ 
14:      end if
15:    end for
16:    if  $matching\_count \geq (issuelist.size/2)$  then
17:       $NeighborhoodBidList.add(AnAgreedBid)$ 
18:    end if
19:  end for
20:  return NeighborhoodBidList
```

【頻度探索によるパレート改善】

頻度探索とは, 探索範囲を相手の提案履歴において頻出する要素に限定する局所探索である。頻度探索は, 相手が頻繁に提案する合意案候補の要素が, 相手の効用関数において重要な制約を充足するというヒューリスティックスに

基づいている．非線形効用関数 [14] の場合，二項制約のように複数の要素によって充足される制約が存在する．したがって，頻出する要素を組み合わせた合意案候補が必ず相手にとって効用値が大きいとは限らない．しかし，非線形効用関数であっても，K-近傍法のみよりも，頻度探索を併用する K-近傍法のほうが，相手に提案する合意案候補のパレート改善が期待できる．理由は，相手の提案履歴において頻出する要素の方が相手がより好んでいる可能性が高いためである．K-近傍法の場合のみは過去に合意したか否かという情報であり，頻度までは考慮されていない．頻度探索による合意案候補の探索は次のような処理で行われる．Algorithm 3 に詳細を示す．

- (1) 基準となる合意案候補を，近傍探索で獲得した合意案候補群 *NeighborhoodBidList* とする (1 行目)．
- (2) 交渉プラットフォームの *Genius* の提供する情報の 1 つである *allValueFrequencyList* を獲得し，各論点で，最も選択された頻度の高い選択肢を獲得する (2 行目)．
- (3) *NeighborhoodBidList* と比較し，選択肢が同じであれば *matchingcount* をインクリメントする (9 行目から 24 行目)．
- (4) *matchingcount* が論点数の半数以上であれば *Frequen-*

Algorithm 3 頻度探索による合意案候補の探索アルゴリズム詳細

```

1: NeighborhoodBidList: 近傍に Offer された合意案と類似度の高い合意案候補群
2: FrequencyBidList: 交渉相手の提案頻度の高い合意案候補群
3: allValueFreqList: 交渉相手の選択肢の頻度行列
4: issueList: 論点のリスト
5: Function getBidbyFrequencySearch (Neighborhood-
   BidList)
6: for  $i = 0$  to NeighborhoodBidList.size do
7:   matchingCount := 0 // NeighborhoodBid と FrequencyBid
     の類似度指標
8:   for  $j = 0$  to issueList.size do
9:     valueList = issueList.getValue // 各論点の選択肢リスト
10:    for  $k = 0$  to valueList.size do
11:      if allValueFreqList[ $k$ ]  $\geq$  highValueCount then
12:        // 各論点の中で最も選択された選択肢の選択回数
13:        highValueCount = allValueFreqList[ $k$ ]
14:        // 各論点の中で最も選択された選択肢
15:        highValue = allValueFreqList.getValue
16:      end if
17:    end for
18:    NeighborValue := NeighborhoodBid に含まれる論点に
      与えられた選択肢
19:    if NeighborValue == highValue then
20:      matchingCount ++
21:    end if
22:  end for
23:  if matchingCount  $\geq$  (issueList.size/2) then
24:    FrequencyBidList.add(NeighborhoodBidList[ $i$ ])
25:  end if
26: end for
27: return FrequencyBidList

```

cyBidList に追加する (25 行目から 27 行目)．

5. 評価実験と考察

5.1 評価実験設定

本論文で提案するエージェントの評価を行うために，ANAC 決勝大会と PRIANAC 決勝大会のエージェントと対戦して，獲得効用スコアと社会的余剰スコア，ナッシュ交渉解との距離 (Distance to Nash)，交渉経過時間 (Time) を導出する．評価方法は ANAC の交渉ドメインを用いて提案エージェントと既存上位エージェントを含めた全 6 エージェントを総当たりで交渉を行う．獲得効用スコア，社会的余剰スコア，ナッシュ交渉解との距離，交渉経過時間は各エージェントが行うすべての交渉の平均を測定して求める．エージェント同士が行う交渉は 10 秒間で 1 回の交渉とし，同じ交渉条件で交渉を 100 回繰り返す．100 回繰り返したらエージェント同士の交渉は終了とし，100 回交渉を行って獲得した獲得効用スコア，社会的余剰スコア，ナッシュ交渉解との距離，交渉経過時間の平均を算出する．K-近傍法において過去のデータの参照は，同一ドメイン，同一対戦相手に対して行う．表 1 は評価実験で使用する ANAC の交渉ドメインのドメインの大きさ，論点数，割引効用 *df*，および留保価格 *RV* をまとめた表である．表 2 は評価実験で使用する ANAC，および PRIANAC に出場したエージェントをまとめた表である．

5.2 評価結果と考察

【概要】

表 3 のグラフから，提案するエージェントは ANAC の

表 1 ANAC で用いられた制約集合

Table 1 Constraint set used in ANAC.

ドメイン名	論点数	ドメインの大きさ	df	RV
partydomain	6	3,072	1.0	0
smartEnergyGrid	4	625	1.0	0
DomainTwf	2	25	0.2	0.05
Bank-robbery	3	18	1.0	0.2
Tram	7	972	1.0	0.5
freemarket	4	625	0.1	0.25
JapanTrip	4	240	0.9	0.9
triangularFight	2	9	0.1	0.3

表 2 過去の上位エージェント

Table 2 Past upper agents in the past.

エージェント名	出場年度
Atlas3	ANAC2015 決勝出場エージェント
Caduceus	ANAC2016 決勝出場エージェント
Farma	ANAC2016 決勝出場エージェント
PonPokoAgent	ANAC2017 決勝出場エージェント
Agent33	PRIANAC 社会余剰部門優勝エージェント

表 3 既存上位エージェントを加えた場合の獲得効用値の平均と社会的余剰の平均（括弧内は標準偏差）

Table 3 Utility and social social welfare when existing superior agent is added.

エージェント名	獲得効用値	順位	社会的余剰	順位
Atlas3	0.839 (0.160)	3	1.345 (0.536)	5
Caduceus	0.782 (0.198)	5	1.320 (0.480)	6
Farma	0.884 (0.097)	1	1.495 (0.439)	2
PonPokoAgent	0.846 (0.178)	2	1.395 (0.524)	3
Agent33	0.696 (0.190)	6	1.395 (0.444)	4
Our Agent	0.802 (0.081)	4	1.522 (0.336)	1

表 4 既存上位エージェントを加えた場合のナッシュ交渉解との距離の平均と交渉時間の平均（括弧内は標準偏差）

Table 4 Distance to Nash and Negotiation time when existing superior agent is added.

エージェント名	ナッシュ交渉解との距離	順位	交渉時間	順位
Atlas3	0.345 (0.355)	6	5.437 (4.106)	5
Caduceus	0.286 (0.307)	4	7.119 (2.529)	6
Farma	0.189 (0.240)	2	5.142 (3.841)	3
PonPokoAgent	0.217 (0.299)	3	5.320 (3.483)	4
Agent33	0.314 (0.275)	5	4.483 (3.848)	2
Our Agent	0.135 (0.174)	1	4.481 (3.289)	1

様々な交渉ドメインでは、既存上位エージェントと比べて社会的余剰値におけるスコア順位が最も大きいことが分かる。Atlas3, Agent33 は ANAC, PRIANAC の社会的余剰部門の優勝エージェントだが、提案エージェントは表 3 のとおり、Atlas3, Agent33 よりも高い社会的余剰値を獲得することができた。提案エージェントの合意案候補の探索手法は交渉相手の *Offer* した時間も参考にしたため、既存上位エージェントよりも安定して高い社会的余剰値を獲得できたと考えられる。

表 4 は既存上位エージェントを加えた場合のナッシュ交渉解との距離（Distance to Nash）と交渉時間（Time）を示している。表 4 から、提案するエージェントは ANAC の様々な交渉ドメインでは、既存上位エージェントと比べて提案エージェントはナッシュ交渉解から最も近い距離で合意しており、合意に至るまでの交渉時間が最も短いことが分かる。提案エージェントは過去の交渉情報を参照し合意案候補の探索を行うため、交渉の序盤から交渉相手の選好情報を予測して合意案候補の探索を行うことができる。ANAC 上位エージェントは現在行っている交渉の情報のみを参照した頻度探索、近傍探索によるパレート改善を行っていた。Farma はナッシュ交渉解との距離が全エージェントの中で 2 番目に近くパレート改善が行えているが、交渉時間は PRIANAC の上位エージェントである Agent33 と提案エージェントよりも遅い。ANAC 上位エージェントの合意案候補の探索手法の問題であった早期合意形成という観点に対して、PRIANAC のエージェントである Agent33 と提案エージェントは ANAC 上位エージェントよりも早

期合意形成を行うことができた。Agent33 は交渉相手の提案応答を次の交渉から参照することにより、交渉の序盤から十分な情報量でパレート改善を行うことが可能であるが、交渉時間を考慮したパレート改善を行わないため合意形成には至るがナッシュ交渉解との距離が遠くなってしまふ。提案エージェントは交渉時間を考慮したことにより、表 4 のとおり早期合意形成、ナッシュ交渉解という観点で既存エージェントよりも優れていることを示した。

【近傍数 K について】

本論文では、近傍数 K を変化させ、ドメインに依存して最適な K がどの程度異なるかについて試行錯誤したが、一般性のある知見を得ることは今の所できていない。これは、ANAC や PRIANAC という競技会の性質として、交渉相手のエージェントおよびドメインの設定によって大きく交渉自体の性質が異なる点があげられる。交渉相手やドメインによって交渉自体の性質が異なることから、一般性を見い出せないため、本論文での K の値の決め方は、K 近傍法において一般的な経験則として知られる「データ数の平方根」を用いた（文献 [14] など）。すなわちここではデータとなる合意案候補の数の平方根として計算している。本手法を用いることで、上記で示すような、既存エージェントを超える、高い獲得効用値、高い社会的余剰値、ナッシュ交渉解の近さ、交渉時間の短さを実現している。

理想的には、K に対して結果がどの程度ロバストか、ドメインに依存した最適な K がどの程度になるかについて調査する必要がある。たとえば、理想的なドメインと理想的な交渉相手を仮定したうえで理想的な K を求め、具体的な

交渉相手やドメインに対応させていくことが可能である。ただし、ANAC や PRIANAC という競技会の性質としてベースラインとなるデータセットが揃っていない状況である。一般的な機械学習に関する研究は、一般的にはベースラインとなるデータセットが十分に揃っている状況で理想的な K を求める。したがって、ANAC や PRIANAC の競技会を対象として行うには、ベースラインとなるより多くのデータセットと、そのデータの整理、そして理想的な K に関する考察と、ANAC や PRIANAC で使われている GENIUS というゲーム環境を用いた多くの実時間を必要とする評価実験を実行する必要がある。これは大変興味深い研究課題であり、今後の課題とする。

【平均値と標準偏差による結果の有意差について】

平均値と標準偏差により各評価項目について以下のよう
に考察する。

- 獲得効用値については、標準偏差が平均値に対して $1/5 \sim 1/10$ となっており、バラツキがやや大きい結果である。
- 社会的余剰値については、標準偏差が平均値に対して $1/2 \sim 1/4$ となっており、バラツキはそれほど大きくない。
- 交渉時間については、どの結果も平均値に標準偏差が非常に近く、バラツキの少ない結果といえる。
- ナッシュ交渉解の近さについては、平均値に対して標準偏差が大きい。これはナッシュ交渉解との近さについては現状のデータ数では正規分布が仮定できないと思われる。

以上より、本論文の目的である、社会的余剰値と交渉時間については、バラツキの少ない結果を得ることができている。

6. おわりに

本研究では、相手の選好情報が明らかでない二者間複数論点交渉問題において K -近傍法と頻度探索、近傍探索による合意案候補の探索を実現した。 K -近傍法による分類を行うことであらかじめ互いに利得を高められるような合意案候補を獲得し、頻度探索と近傍探索によるパレート改善を行うことで交渉ドメインに依存することなく社会的余剰値を高めた。また過去交渉情報を参照することにより早期合意形成も可能にした。今後の課題として、より多くの交渉問題で提案するエージェントの性能を評価することや、提案するエージェントの合意案候補探索手法を多者間交渉問題へ拡張することなどがあげられる。さらに、実験では両者を総合した評価なので、どちらの方がより効いているのか、またドメインによって効果の傾向が変わるのか、実験および考察も今後の課題とする。

参考文献

- [1] Baarslag, T.: What to bid and when to stop, PhD Thesis, Delft University of Technology (2014).
- [2] Baarslag, T., Fujita, K., Gerding, E.H., Hindriks, K., Ito, T., Jennings, N.R., Jonker, C., Kraus, S., Lin, R., Robu, V., et al.: Evaluating practical negotiating agents: Results and analysis of the 2011 international competition, *Artificial Intelligence*, Vol.198, pp.73–103 (2013).
- [3] Brzostowski, J. and Kowalczyk, R.: Adaptive negotiation with on-line prediction of opponent behaviour in agent-based negotiations, *Proc. IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pp.263–269, IEEE Computer Society (2006).
- [4] Chen, S. and Weiss, G.: An Efficient and Adaptive Approach to Negotiation in Complex Environments, *ECAI*, Vol.242, pp.228–233 (2012).
- [5] Faratin, P., Sierra, C. and Jennings, N.R.: Negotiation decision functions for autonomous agents, *Robotics and Autonomous Systems*, Vol.24, No.3-4, pp.159–182 (1998).
- [6] Hassanat, A.B., Abbadi, M.A., Altarawneh, G.A. and Alhasanat, A.A.: Solving the problem of the K parameter in the KNN classifier using an ensemble learning approach, arXiv preprint arXiv:1409.0919 (2014).
- [7] Ito, T., Hattori, H. and Klein, M.: Multi-issue Negotiation Protocol for Agents: Exploring Nonlinear Utility Spaces, *IJCAI*, Vol.7, pp.1347–1352 (2007).
- [8] Iwasa, K. and Fujita, K.: Prediction of Nash Bargaining Solution in Negotiation Dialogue, *Pacific Rim International Conference on Artificial Intelligence*, pp.786–796, Springer (2018).
- [9] Kawaguchi, S., Fujita, K. and Ito, T.: AgentK: Compromising strategy based on estimated maximum utility for automated negotiating agents, *New Trends in Agent-Based Complex Automated Negotiations*, pp.137–144, Springer (2012).
- [10] Liu, S., Moustafa, A. and Ito, T.: Agent33: An Automated Negotiator with Heuristic Method for Searching Bids Around Nash Bargaining Solution, *International Conference on Principles and Practice of Multi-Agent Systems*, pp.519–526, Springer (2018).
- [11] Luo, X., Jennings, N.R., Shadbolt, N., Leung, H. and Lee, J.H.: A fuzzy constraint based model for bilateral, multi-issue negotiations in semi-competitive environments, *Artificial Intelligence*, Vol.148, No.1-2, pp.53–102 (2003).
- [12] Luo, X., Leung, H. and Lee, J.H.: Theory and properties of a selfish protocol for multi-agent meeting scheduling using fuzzy constraints, *ECAI*, pp.373–377 (2000).
- [13] Matsune, T. and Fujita, K.: Weighting estimation methods for opponents' utility functions using boosting in multi-time negotiations, *IEICE Trans. Information and Systems*, Vol.101, No.10, pp.2474–2484 (2018).
- [14] Osborne, M.J. and Rubinstein, A.: *Bargaining and markets*, Academic Press (1990).
- [15] Rubinstein, A.: Perfect equilibrium in a bargaining model, *Econometrica: Journal of the Econometric Society*, pp.97–109 (1982).
- [16] Williams, C.R., Robu, V., Gerding, E.H. and Jennings, N.R.: Using gaussian processes to optimise concession in complex negotiations against unknown opponents, *22nd International Joint Conference on Artificial Intelligence* (2011).
- [17] 藤田桂英, 森 顕之, 伊藤孝行ほか: ANAC: Automated Negotiating Agents Competition (国際自動交渉エージェ

ント競技会) (特集) 人工知能研究のベンチマークとは—標準問題・データセット・評価手法), 人工知能, Vol.31, No.2, pp.237–247 (2016).



奥原 俊 (学生会員)

2009 年名古屋工業大学電気情報学部卒業。2013 年同大学大学院博士前期課程産業戦略工学専攻修了。2013 年同大学大学院博士前期課程工学専攻科情報工学専攻。2015 年藤田保健衛生大学 (現、藤田医科大学) 医療科学部医療経営情報学科助教。現在に至る。

濱田 大槻 (学生会員)

2019 年名古屋工業大学工学創成プログラム学科卒業。現在に至る。

アーメッド ムスタファ

ウーロンゴン大学博士課程修了。アデレード大学客員研究員、オークランド工科大学客員研究員、および Data61 客員研究員を経て名古屋工業大学准教授。現在に至る。自動交渉、マルチエージェント強化学習、マルチエージェント社会における信頼と評判、深層強化学習、サービス指向コンピューティング、集合知、知能交通システム、データマイニングに興味がある。ICWS, ICSOC, および WWW 運営審査委員。人工知能学会, IEEE Computer Society, Australia Computer Society, および Service Science Society of Australia 各会員。



伊藤 孝行 (正会員)

2000 年名古屋工業大学大学院博士後期課程修了。博士 (工学)。1999 年 JSPS 特別研究員。2000 年 USC/ISI 客員研究員。2001 年北陸先端科学技術大学院大学助教。2003 年名古屋工業大学大学院助教。2005 年ハーバード大学および MIT 客員研究員。2006 年名古屋工業大学大学院産業戦略工学専攻准教授。2008 年 MIT 客員研究員。2009 年 JST さきがけ大挑戦型研究員。2014 年名古屋工業大学大学院産業戦略工学専攻/情報工学教育類教授。現在に至る。2011 年内閣府最先端・次世代研究開発プロジェクト代表研究者。2015 年名古屋工業大学大学院産業戦略工学専攻専攻長。2015 年 JST CREST 研究代表者。2014 年日本ソフトウェア科学会基礎研究賞。2014 年日本学術振興会賞受賞。2013 年文部科学大臣表彰科学技術賞受賞。2007 年文部科学大臣表彰若手科学者賞受賞。情報処理学会長尾真記念特別賞受賞。2006 年 AAMAS2006 最優秀論文賞受賞。2005 年日本ソフトウェア科学会論文賞受賞。平成 16 年度 IPA 未踏ソフトウェア創造事業スーパークリエータ認定。AAMAS2013 Program Chair。マルチエージェントシステム国際財団 (IFAAMAS) 理事。