

発想的議論を用いた譲歩と妥協の形式化に基づくエージェント間交渉

関口 知之[†] 若木 利子[‡]

芝浦工業大学大学院 工学研究科 電気電子情報工学専攻[†]

芝浦工業大学 システム理工学部 電子情報システム学科[‡]

1 はじめに

近年、電子商取引引き等の複数人間の議論に基づく意思決定や問題解決などの行為における「交渉、譲歩、妥協、合意」などのメカニズムを形式化しコンピュータネットワーク上のマルチエージェントにシミュレートさせる研究が進展している [1] [3] [5]。最近、和歌山大学の坂間 [2] がエージェントの知識を ELP/EDP の論理プログラムで記述し、解集合プログラミング (ASP) とその意味論に基づくインダクション・アブダクションを用いた譲歩、妥協の形式化を行った。人間の行う議論においては本質的に矛盾した知識を扱うが、ASP に基づく方法では知識が無矛盾である場合にしか適用できない。他方、議論の意味論では一貫性制約 (integrity constraint) が扱えなかった。本研究では矛盾した知識や一貫性制約を扱える議論の意味論や、議論でアブダクションを可能にする発想的議論 [4] を適用し、交渉における譲歩、妥協の形式化を行い、提案手法に基づくシステムの実装と解の評価を行った。

2 準備

2.1 エージェントの議論

議論とは意見を述べ合い、どの意見が勝利するかを判定する行為である。プリファレンスを扱う議論の意味論 [3, 6] を示す。

定義 1 (拡張論理プログラム) [3] リテラル L は原子式 A 、またはその論理否定 $\neg A$ である。not はデフォルト否定であり、not A を NAF リテラルと呼ぶ。拡張論理プログラム (ELP) は以下の形式のルール (r) の集合である。

$$L_0 \leftarrow L_1, \dots, L_m, \text{not } L_{m+1}, \dots, \text{not } L_{m+n} \quad (1)$$

(但し $m, n \geq 0$)。← に対して左辺を頭部、右辺を本体と呼び、それぞれ $\text{head}(r)$, $\text{body}(r)$ で表す。

定義 2 (論証と攻撃関係) [3] ELP P の論証 Ag とは、 $r_i \in P$ の有限系列 $Ag = [r_1, \dots, r_n]$ であり、 $\forall L \in \text{body}(r_i)$ ($1 \leq i \leq n$) に対して $\text{head}(r_k) = L$ なる $k > i$ が存在しなければならない。論証 Ag に含まれているルールの頭部を結論、NAF リテラルを仮定という。P に関する極小論証の集合を Args_P , $\text{head}(r_1)$ を Ag の主張として $\text{claim}(Ag)$ で表す。

攻撃関係 R は Args_P 上の 2 項関係 ($R \subseteq \text{Args}_P^2$) で定義され、*rebut*, *undercut*, *attack*, *defeat* 等がある。

定義 3 (ルール間優先関係と論証間優先関係) ELP P 上の優先関係 (プリファレンス) を定めるものとしてルール間優先関係 \leq を定義し、 \leq が与えられたとき、 Args_P 上の優先関係として論証間優先関係 \sqsubseteq を文献 [6] に従って定義する。ここで $Ag_1 \sqsubseteq Ag_2$ は「論証 Ag_1 より論証 Ag_2 を優先する」ことを意味する。

定義 4 (議論フレームワーク) 所与の ELP P と P 上のルール間優先関係 \leq より議論フレームワーク AF_P を、

$AF_P \stackrel{\text{def}}{=} (\text{Args}_P, \text{def}_P)$ として定義する。 def_P は「優先関係を考慮した攻撃関係」で $\text{def}_P \stackrel{\text{def}}{=} R - \sqsubseteq$ で定義する。

定義 5 (議論の意味論) [1] $S \subseteq \text{Args}_P$ を *conflict free* な論証集合、単調関数 $F: 2^{\text{Args}_P} \rightarrow 2^{\text{Args}_P}$ を $F(S) = \{A \mid \text{論証 } A \text{ は } S \text{ に関して受理可能}\}$ と定義したとき、 AF_P における *complete*, *preferred*, *grounded*, *stable* の議論の意味論は以下で定義される *extension* E の集合で与えられる。

- E が *complete extension* iff $E = F(E)$.
- E が *preferred extension* (*grounded extension*) iff E は \sqsubseteq に関する極大 (極小) な *complete extension*.
- E が *stable extension* iff *preferred extension* である E について、 $\text{Args}_P \setminus E$ の任意の論証を攻撃する論証が E に存在。

Argumentation-based negotiation based on formalising of concession and compromise applying Abductive Argumentation.

[†]Tomoyuki Sekiguchi, [‡]Toshiko Wakaki

[†]Graduate School of Engineering, Shibaura Institute of Technology Department of Electrical Engineering and Computer Science

[‡]Shibaura Institute of Technology Department of Electronic Information Systems

2.2 発想的議論

ELP P と仮説集合 H の対である発想的プログラム $\langle P, H \rangle$, 及び議題 G が与えられて、 $E \subset H$ なる E と P より作られる論証集合 $\text{Args}_{P \cup E}$ とその攻撃関係 def において、 G を結論に持つ論証 $Ag \in \text{Args}_{P \cup E}$ が *skeptical* (或は *credulous*) に正当化されるような説明 E を求めるのが発想的議論である。発想的議論の詳細は [4] を参照。(なお G 及び仮説は P の言語のリテラルである。)

3 制約付き議論フレームワーク

本研究で提案する一貫性制約を扱える議論の意味論を示す。

定義 6 (制約付き議論フレームワーク) (1) の形式のルールの集合を P , 次の (2) の形式のルール、

$$\leftarrow L_1, \dots, L_m, \text{not } L_{m+1}, \dots, \text{not } L_{m+n} \quad (m, n \geq 0) \quad (2)$$

即ち一貫性制約の集合を C , C の各ルールに予約リテラル \perp を頭部に追加して (1) の形式にしたルールの集合を P_C とする。このとき、ELP $P \cup C$ に関する制約付き議論フレームワーク: $CAF(P, C)$ の意味論は、ELP $K = P \cup P_C$ に関する議論フレームワーク AF_K を用いて、以下で定義される C -extension の集合で与えられる。

- E に属する任意の論証の *claim* が \perp でないとき、 $E \subseteq \text{Args}_K$ が C を満たす。
- 指定の議論意味論の下で E が $CAF(P, C)$ の C -extension iff E が AF_K の指定の議論意味論の *extension* であり、かつ C を満たす。

定義 7 (Skeptical/Credulous に正当化) 制約付き議論フレームワーク $CAF(P, C)$ における E_i^C ($i \geq 0$) を指定の議論意味論の C -extension とする。論証 $Ag \in \text{Args}_{P \cup P_C}$ について「全ての (或る) E_i^C について $Ag \in E_i^C$ ならば、 Ag は *skeptical* (*credulous*) に正当化される。」

4 マルチエージェントの交渉

本研究では、2 エージェントを想定し、各エージェントは発想的プログラム $\langle P \cup C, H \rangle$ で表現された知識ベースを持つ。P は信念集合、C は一貫性制約の集合、H は仮説集合である。但し、P を強い信念集合 Σ と状況により棄却できる弱い信念集合 Γ に分割して、 $P = \Sigma \cup \Gamma$ とする。

本研究の譲歩と妥協の形式化の基本的なアイデアを述べる。相手からの提案 G に対して、 $CAF(P, C)$ に対して *skeptical* に正当化するのが望ましいが、① *credulous* に正当化されれば良しとする、② G を *credulous* に受け入れるための条件を発想的議論で求めて相手に要求する、③ G を受け入れるために自身の信念やプリファレンスを棄却する、④ ②と③の組み合わせ、といった段階的な譲歩の方法を考えた。妥協は、 G は受け入れられないが、それに近い受理できる提案を G の一般化により発見し、それを提示するものとしてその方法を以下に提案する。

<< 提案手法 >>

(エージェントの提案の判定) エージェントは相手が提示した提案 G (但し、 G はリテラル) を以下のように判定する。

- $\text{claim}(Ag) = G$ なる Ag が $CAF(P, C)$ に関して指定の議論意味論で *skeptical* に正当化されるならば、「 G は *accepted*」という。
- 指定の議論意味論の C -extension が存在しないならば、「 G は *unacceptable*」という。
- それ以外の場合ならば「 G は *acceptable*」という。

次に提案 G が *acceptable* であるならば、エージェントは譲歩によって受理できるかの判定、妥協の提示を以下に従って行う。(譲歩と妥協) 譲歩には 4 つのタイプが存在する。

- **Type1** *credulous* な正当化による譲歩。
 $\text{claim}(Ag) = G$ なる Ag が $CAF(P, C)$ に関して指定の議論意味論で *credulous* に正当化される。
- **Type2** 条件 E の提示による譲歩。
 $\text{claim}(Ag) = G$ なる Ag が $CAF(P \cup E, C)$ に関して指定の議論意味論で *credulous* に正当化される。条件 E は発想的議論を用いて求める。
- **Type3** 信念 J の棄却による譲歩。
 $\text{claim}(Ag) = G$ なる Ag が $CAF(P \setminus J, C)$ に関して指定の議論意味論で *credulous* に正当化される。

• **Type4** Type2 と Type3 の組み合わせ.

$claim(Ag) = G$ なる Ag が $CAF((P \setminus J) \cup E, C)$ に関して指定の議論意味論で credulous に正当化される。いずれの譲歩においても提案 G が受理できない場合, Type1, 2, 3, 4 のいずれかを満たす新たな提案 G' を発見し, 相手に提示することを本研究では妥協と呼ぶ。 G' の発見には一般化 (generalization) を利用する。

次に上記の提案手法を実現した交渉するマルチエージェントシステムの概要を述べる。 エージェントは自身の信念に基づく議論によって交渉動作 (move) の可否を決定し, 相手に可能な動作を提示することで交渉を行う。(交渉動作:move) move とはエージェントが交渉の際に相手に提示する動作であり 8 種類存在する。これにより提案の提示, 拒否等を行う。 move の一部の詳細を表 1 に示す。

表 1 move の種類

move	question(G(X))	offer(G)	offer(E)
発行条件	G(X) は入力。	G は自身の知識から妥協 Type1, Type3 で発見。即ち G は自身で受理可能。	相手の提案が譲歩 Type2 Type4 で受理可能。
返答可能な動作	offer(G)	accept(G), refuse(G) offer(G'), offer(E)	accept(E), refuse(E) offer(E')
move	refuse(G)	refuse(E)	accept(G)
発行条件	G が unacceptable である。	unacceptable である $e \in E$ なる e が存在する。	G が accepted, 又は譲歩 Type1, Type3 で受理可能。
返答可能な動作	offer(G')	offer(G), offer(E')	交渉停止

システム構成 本研究のマルチエージェントシステムは, Log の管理 (図 1.(3)) やクライアントに move 提示 (図 1.(1),(5)) を行うサーバと, エージェントとして返答 move 作成 (図 1.(2)) の検討を行うクライアントから構成する。



図 1 交渉するマルチエージェントシステムの構成図

サーバは move をノードとする交渉木をスタック S を用いて構築・制御を行い, それに基づいて深さ優先探索を行う。その経路より得られる $question(G(X)) \sim accept(G)$ までの動作系列が交渉の解となる。また, バックトラックを行っていくことにより, 解となる全経路を求めることが可能である。より詳細なシステムの手続きの概要を以下に示す。

- Step1 クライアントは相手への質問がある場合, 質問 $G(X)$ を入力し, サーバへ送信する。
- Step2 サーバは他方のエージェントに $question(G(X))$ を提示する。
- Step3 クライアントは提示された move に対して可能な全ての返答 move をリストとしてサーバへ送信し, サーバはそのリストを S に push する。
- Step4 サーバは S を pop し, その move が $accept(G)$ ならば Step5 へ。そうでなければその move をクライアントに提示し Step3 へ。 S が空であるなら Step6 へ。
- Step5 交渉合意と判断し, クライアントに合意経路を送信する。交渉を継続するならば Step4 へ。
- Step6 一度でも Step5 に進んでいるならば両クライアントへ全合意経路を送信し, そうでなければ交渉決裂を送信して交渉終了。

文献 [2] example3.2 で示された交渉の例題を用いて, 本システムにおける交渉の計算過程を説明する。

例題 HDD を買いたいエージェント B と売り手のエージェント S がいる。両エージェント間で行われる交渉の流れを以下に示す。(ここでは preferred 意味論で判定する。)

- B_1 : 200G の HDD はいくらですか。
- S_1 : それは 120\$ です。
- B_2 : 私はそれを 100\$ でほしいのですが。
- S_2 : 現金で支払うなら, 100\$ に値引きします。
- B_3 : 現金では支払えません。
- S_3 : 180G の HDD ならば, 100\$ でお売ります。
- B_4 : それでお願いします。

買い手 B , 売り手 S の知識ベースをそれぞれ表 2, 表 3 に示す。表 2 の $\leq (b2, b1)$, 表 3 の $\leq (s2, s1)$ は引用例題に追加した知識であり, $\leq (b2, b1)$ は “ルール $b2$ よりルール $b1$ を優先する” ことを意味する。

提案手法に基づき, 例題の交渉の流れを得るプロセスを以下に示す。ここでは B と S の毎回のやり取りの対を round として扱い, $P_B = \cup_{B} P_B, P_S = \cup_{S} P_S$ とする。

表 2 買い手 B の知識ベース

Σ_B	$b1 : product(hdd, 200G, 100\$) \leftarrow not \neg product(hdd, 200G, 100\$).$ $b2 : product(hdd, 180G, 100\$) \leftarrow not product(hdd, 200G, 100\$).$ $\neg product(hdd, 200G, 100\$) \leftarrow product(hdd, 180G, 100\$).$ $high_price \leftarrow product(hdd, 200G, Y), Y > 100.$
Γ_B	$\leq (b2, b1).$
C_B	$\leftarrow pay_cash. \quad \leftarrow high_price.$
H_B	

表 3 売り手 S の知識ベース

Σ_S	$s1 : product(hdd, 200G, 120\$) \leftarrow not \neg product(hdd, 200G, 120\$).$ $product(hdd, 180G, 100\$) \leftarrow not \neg product(hdd, 180G, 100\$).$ $s2 : product(hdd, 200G, 100\$) \leftarrow pay_cash.$ $\neg product(hdd, 200G, 120\$) \leftarrow product(hdd, 200G, 100\$).$
Γ_S	$\leq (s2, s1).$
C_S	
H_S	$pay_cash.$

(1st round) B が S に 「200G の HDD はいくらか」を以下の move を用いて質問することで交渉が開始される。

$B_1 : question(product(hdd, 200G, X))$

S は B_1 への返答として $CAF(P_B, C_S)$ において accepted である提案 $product(hdd, 200G, 120\$)$ を B へ提案する。

$S_1 : offer(product(hdd, 200G, 120\$))$

(2nd round) B にとって S_1 の提案は, unacceptable である。よって受理できず, B は $CAF(P_B, C_B)$ において妥協 Type1 で受理される提案 $product(hdd, 200G, 100\$)$ を妥協案として提示する。 $B_2 : offer(product(hdd, 200G, 100\$))$ 。

S にとって提案 $product(hdd, 200G, 100\$)$ は acceptable である。よって譲歩を行うと Type2 における $E = \{pay_cash\}$ の場合に受理できると判定する。よって E を条件として提示する。 $S_2 : offer(\{pay_cash\})$ 。

(3rd round) B にとって S_2 の提案は unacceptable であるため, これを拒否する。 $B_3 : refuse(\{pay_cash\})$ 。

S は条件が拒否されたため, $product(hdd, 200G, 100\$)$ を受理する他の方法を探すが発見できない。そこで妥協 Type1 で受理される提案 $product(hdd, 180G, 100\$)$ を妥協案として提示する。 $S_3 : offer(product(hdd, 180G, 100\$))$ 。

(4th round) B にとって提案 $product(hdd, 180G, 100\$)$ は acceptable である。よって譲歩を行うと Type3 における $J = \{\leq (b2, b1)\}$ の場合に受理できると判定し, この提案を受け入れる。 $B_4 : accept(product(hdd, 180G, 100\$))$ 。

5 実装と評価

交渉を行うマルチエージェントシステムは, Java を用いて実装した。クライアントとサーバは socket による通信を介してやり取りを行う。複数の例題を用いて, 定義に基づく計算結果と, 本システムの実行結果が一致していることを確認し, システムの正当性を検証した。更に, 得られる複数の解に対してパレート最適等を用いた解の評価の方法の検討を行った。以下の図は例題を実際にシステムに計算させた際の画面であり, 図 2 はクライアント画面, 図 3 は合意で終了した全動作系列を表示する結果画面である。

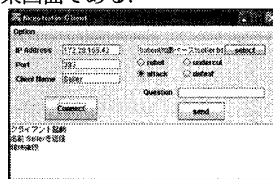


図 2 クライアント画面

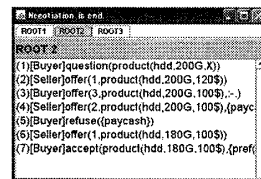


図 3 結果画面 (全合意経路表示)

6 終わりに

解を探索するプロセスにおいて最適な解を優先的に発見する様な探索制御の方法の提案, ILP の種々の技法を検討し, より高度な人間に近い妥協を実現することが今後の研究課題である。

参考文献

- [1] P. M. Dung: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, Artificial Intelligence, 77, pp.321-357, 1995.
- [2] C. Sakama: Inductive Negotiation in Answer Set Programming, 6th International Workshop on Declarative Agent Languages and Technologies, LNAI 5397, Springer-Verlag, pages 143-160, 2008.
- [3] R. Schweimeier, M. Schroeder: A Parameterised Hierarchy of Argumentation Semantics for Extended Logic Programming and its Application to the Well-founded Semantics, TPLP, pp.207-242, 2005.
- [4] T. Wakaki, K. Nitta, H. Sawamura: Computing Abductive Arguments in Answer Set Programming, Proc. of ArgMAS 2009, pp.189-206, 2009
- [5] 関口知之, 大久保崇行, 若木利子: 議論に基づく交渉システムの実装, 情報処理学会第 71 回全国大会, 2R-6, 2009.
- [6] 関口知之, 若木利子: 動的プリファレンスを扱う議論エージェントシステムの構築, 電子情報通信学会 (信学技報), AI2008-3, pages 15-20, 2008.