社団法人　情報処理学会　研究報告
IPSJ SIG Technical Reports

2005－ＩＣＳ－139（11）
2005／3／15

投機的に計算するマルチエージェントシステムにおける実時間意思決定に関する考察

岩内　栄二[†,††]　井上　克巳[††]

† 神戸大学大学院自然科学研究科　〒658-8501 神戸市灘区六甲台町 1-1
†† 国立情報学研究所　〒101-8430　東京都千代田区一ツ橋 2-1-2
E-mail: †034t209n@y03.kobe-u.ac.jp, ††ki@nii.ac.jp

あらまし　本論文では，不完全な通信環境下での投機的計算を用いた論理型マルチエージェントシステムにおける意思決定問題について考察する．本研究での意思決定問題とは，投機的計算によって得られた解をどのタイミングで実際の行動に移すかを決めることである．このために，期待効用理論に基づきエージェントに対する効用を定めることにより，実時間で合理的にエージェントの意思を決定するアルゴリズムを提案する．

キーワード　実時間意思決定，投機的計算，論理型マルチエージェントシステム

Real-time decision-making in multi-agent systems which perform speculative computation

Eiji IWAUCHI[†,††] and Katsumi INOUE[††]

† Graduate School of Science and Techology, Kobe University
1-1 Rokkoudai-cho, Nada-ku, Kobe, 658-8501 Japan
†† National Institute of Informatics　2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430 Japan
E-mail: †034t209n@y03.kobe-u.ac.jp, ††ki@nii.ac.jp

Abstract　This paper considers the decision-making problem of logic based multi-agent systems which perform speculative computation under imcomplete communication environments. In this paper, we consider a decision which decides to which timing a solution obtained by speculative computation is put into an actual action. We propose an algorithm to determine an agent's intention rationally in real time by defining a utility to each agent based on the expected utility theory.

Key words　real-time decision-making, speculative computation, logic based multi-agent system

1. はじめに

マルチエージェントシステムとは，複数のエージェントの自律的かつ協調的な行動によって問題解決を実現するシステムである．複数のエージェントによって協調的な行動を行うためには，エージェント間で互いの情報をやり取りするための通信が必要となるが，その通信が必ずしも保証されていない場合がある．その要因としては，通信の際にロストしたり大幅な遅延が起きるなどの通信路による要因，回答をすぐには返さずに自分が有利になるように保持しておくといったエージェントの持つ性質による要因などが挙げられる．そのため，マルチエージェントシステムにおいてはエージェント間のメッセージのやり取りが遅れることを考慮する必要がある．一方，あるエージェントが他のエージェントからの回答に基づいて何らかのプランを立て実行する場合には，プラン決定の遅れが実行の効果を失う場合がある．例えば，ホテルを予約する場合には，どの部屋に泊まるかというプランを決定するのが遅いと予約 (実行) が不可能になる場合がある．すなわち，与えられた問題の締切時間まで他のエージェントからの回答を待ち続けていると協調的な行動の効果が失われてしまうことが考えられる．しかし，多くのマルチエージェントシステムではエージェントが他のエージェントに対し質問を行った場合には，そのエージェントは回答が届いてから処理を行っている．そのため，回答をあらかじめ予想して先行的に行動することは難しく，回答が得られるまでそのまま締切まで待機することになってしまう．

このような問題を解決するための一つの手法が**投機的計算** [4] である．これは，あらかじめデフォルトの仮回答を用意しておき，それを用いて先行的に行動を起こすことである．しかし，

現在までに提案されている投機的計算を用いたマルチエージェントシステム ([2], [7]) では，与えられた問題に対し投機的計算を用いて得られた結論 (プラン) を実行するタイミングについてはあまり議論されていない．文献[7] では，会議室予約問題に対して事前の参加確率を用いて投機的計算を行うか否かを判断しているが，投機的計算を行う場合は直ちに実行している．

そこで，本研究では，投機的計算を行って得られたプランをどのタイミングで実行するのか，というエージェントの意思決定に焦点を当て，期待効用理論に基づきエージェントに効用を与えることにより，実時間で合理的にエージェントの意思を決定することができるシステムを提案する．また，本研究において問題解決には推論エンジンとして **SOL 導出** [1] を使用する．

2. マルチエージェントシステム

本章では，投機的計算の概要の説明と，本研究におけるマルチエージェントシステムの論理的な枠組みを，文献[2] に基づいて説明する．

2.1 投機的計算 (Speculative computation) [4]

投機的計算の概要は以下の通りである．

(1) エージェント A は事前に質問に対するデフォルトの仮回答を用意しておく．

(2) A が他のエージェント B に質問を行った場合に，ただちに，または，回答の〆切になったときに，A は，デフォルトの仮回答を用いて以下の処理を始める．

(3) もし B からの回答が返されたとき，

- 回答が，デフォルトの仮回答と同じであれば，A は計算を続ける．
- もし回答が，デフォルトの仮回答と違っていれば，A は計算を中止し，デフォルトの回答を使う以前の直前の状態から得られた真の回答に基づいて処理をやり直す．

デフォルトの仮回答を用いて先行処理を行った結果は必ずしも正しいものとは言えず，場合によっては再処理の必要があるため本研究ではこのような処理を**投機的計算**と呼ぶ．

投機的計算を行うためにはデフォルトの仮回答が必要となるため，次に示すエージェントの枠組の中にデフォルトを定義する．

2.2 マルチエージェントシステムの枠組み

本研究におけるマルチエージェントシステムの理論的枠組は次の通りである．

[定義 2.1] (マルチエージェントシステム MS)

n エージェントのマルチエージェントシステムの枠組を，2 つ組 $MS = (\Sigma, AF^i)$ で定義する．

- Σ は項の集合でその各要素 i を**エージェント識別子**という．本システム内にある全てのエージェントを表す．
- AF^i はエージェント i の枠組を意味し，次に定義する．

[定義 2.2] (エージェントの枠組 AF, Agent Framework)

エージェントの枠組を，3 つ組 $AF = (\Delta, D, P)$ で定義する．

- Δ はリテラルの集合で，**質問集合**という．その各リテラルの述語を**外部述語**といい，その引数にはエージェント識別子を必ず持つものとする．質問集合の要素を**質問可能リテラル**といい，実際に他のエージェントに対し行う質問を表す．さらに，Δ は Δ_D と Δ_U の二つに分割される．

$$\Delta = \Delta_D \cup \Delta_U, \quad where\ \Delta_D \cap \Delta_U = \phi$$

$L \in \Delta_D$ が成立するリテラル L はデフォルト集合 D でデフォルトの解として定義され，$L \in \Delta_U$ が成立するリテラル L は D でデフォルトの解として定義されない．このため解の完全性は損なわれる．そこで $L \in \Delta_D$ が成立する不確実なリテラル L に対して，その完全性を補う $\neg L$ を共に用いて先行計算行うことによって，起こりうるプランを複数求める．

- D は (部分的な) **デフォルト集合**といい，リテラルの集合である．その各要素は，$L \in \Delta$ または $\neg L \in \Delta$ のいずれかが成立するようなリテラル L である．
- P は節集合で**プログラム**という．ここで，各節は，

$$L_1 \vee \cdots \vee L_n$$

で表され，各 $L_i (i = 1, \ldots, n)$ はリテラルであり，質問集合の要素を含んでいてもよい．

エージェントが問題解決を行う際には，他のエージェントからの返答が得られるたびに信念を更新しなければならない．以下，**時点集合** T は，全順序関係が定義された加算無限集合とし，t_0 を最小の要素とし，任意の時点 $t_i \in T$ に対し次の時点 t_{i+1} が存在するものとする．以下では，返答集合と暫定答集合を時点を含めて定義する．

[定義 2.3] (返答集合, Reply Set) [2]

ある時点 t_i における返答集合 R_i は L または $\neg L$ の形で表されるリテラルの集合であり，各要素リテラルの述語は質問集合 Δ における外部述語である．ただし，質問集合 Δ の各リテラル L に対し，$L \in R_i$ および $\neg L \in R_i$ の両方が成立することはないものとする．返答集合は，他のエージェントからの最新の返答を蓄えておくために使用するものとする．

時点 t_i における返答集合 を R_i とすると，この時点でのエージェントの枠組の意味は次の正規デフォルト理論で表される．

$$\left(\left\{ \frac{:\ L}{L} \;\middle|\; L \in D \right\}, P \cup R_i \right). \qquad (1)$$

このデフォルト理論は常に拡張を持っている[5]．また，(1) の拡張は時点 t_i におけるエージェントの信念集合 B_i を表している．

[定義 2.4] (暫定回答集合, Tentative Answer Set) [2]

時点 t_i における暫定回答集合 T_{R_i} はリテラルの集合であり，t_i における返答集合 R_i と未返答のリテラルに関するデフォルトの和集合で表される:

$$T_{R_i} = R_i \cup \{L \in D \mid L \notin R_i \ and\ \neg L \notin R_i\}.$$

[定理 2.1] (デフォルト理論の拡張) [2]

エージェントの枠組を (Δ, D, P) とし，ある時点 t_i での回答集合と暫定回答集合をそれぞれ R_i, T_{R_i} とする．いま，$P \cup T_{R_i}$ が無矛盾であると仮定する．このとき，論理式の集合 E が正規デフォルト理論 (1) の拡張であることの必要十分条件は，$E = Th(P \cup T_{R_i})$ である．ただし，$Th(\Sigma)$ は Σ の論理的帰結の集合を表す．

定理 2.1 は，デフォルトを用いた投機的計算が，P と暫定回答集合 T_{R_i} の和集合 $P \cup T_{R_i}$ の論理的帰結を計算することと同値になることを示している．このような論理的帰結の計算を行うために，本研究では結論発見について健全かつ完全な手続きである SOL 導出 [1] を用いる．

3. エージェントの意思決定

エージェントは，投機的計算と SOL 導出により現在持っている知識でプランを得ることができる．これまでの研究では SOL 導出により得られたプランをどのタイミングで実行するか議論されていなかった．そこで，本研究では，期待効用理論を用いて利得を定めることにより実時間で判断を決定できるアルゴリズムを提案する．本章では，まず期待効用理論について述べた後，本研究に適用する．

3.1 期待効用理論

期待効用理論 [6] は，偶然性を含む意思決定問題の解決を狙いとしており．その説明にはしばしば“くじ”が用いられる．**くじ** (*lottery*) とは，偶然性を含む選択対象のことであり，

$$l = [x_1, \ldots, x_r; p_1, \ldots, p_r],$$
$$p_1 + \cdots + p_r = 1, p_i \geqq 0, \cdots, p_r \geqq 0$$

で表す．ここで，各 x_i はくじの結果であり，p_i は結果 x_i の生起確率である．とくに，各結果 x_i が賞金であるときくじ l を**賞金くじ**という．一般に，賞金額 x 円に対する意思決定主体の評価を数値化した尺度 $u(x)$ を貨幣に対する**効用** (utilitiy) と言う．また，賞金の期待値を**期待効用** (expected utility) といい次式により求める．

$$\sum_{i=1}^{r} p_i \cdot u(x_i)$$

このとき，賞金くじにおける**期待効用仮説**を，“意思決定主体は複数の賞金くじ l からどの賞金くじを選択するのかという意思決定において，期待効用を最大にする賞金くじを選択する”と表現することができる．本研究では，意思決定を行うタイミングを決定するという実時間における意思決定問題に対して，時間軸上に並んだ複数の“くじ”の中から一つの“くじ”を選択する問題と捉える．そしてエージェントは，期待効用が高い“くじ”を選択すると言う期待効用仮説を満足するもとのして，実時間で合理的にエージェントの意思を決定するアルゴリズムを提案する．

3.2 本研究への適用

本研究における意思決定とは，デフォルトの仮回答を用い投機的計算によって得られた結論 (プラン) を実際の行動に移すかどうか決めることを指している．ここで，本論文ではプランを実際の行動に移すことを“コミットする”，移さないことを“コミットしない”と呼ぶことにする．そして，実行に移すかどうかを決める“くじ”を定義することにより，実行するタイミングを決めるという意思決定問題を，実時間軸上に並べられた意思を決定するための“くじ”を選択する問題と捉える．

3.2.1 エージェントの意思決定

期待効用理論では，結果の集合 X とその要素がそれぞれが生起する確率の集合 P が必要となる．本研究では，“くじ”の選択は“コミットする”か“コミットしない”かであるので，コミットすることを x，コミットしないことを y とすると結果の組 X は，$X = \{x, y\}$ となる．このとき，時点 t_i において，コミットして得られる効用を $u(x(t_i))$，コミットしないことにより得られる効用を $u(y(t_i))$，“コミットする”を選択する確率を $p(t_i)$ とすると，“コミットしない”を選択する確率は $(1-p(t_i))$ であるから時点 t_i における確率の組 P_i は

$$P_i = \{p(t_i), 1 - p(t_i)\}, (0 \leqq p(t_i) \leqq 1)$$

となる．ここで，確率 $p(t_i)$ はデフォルトの回答がどれだけ正しいのかによって変化するものとし，デフォルトの回答の確からしさによって求めることとする．そこで，エージェントの枠組に以下の集合を新たに定義する．

[定義 3.1] **時点 t_i におけるデフォルトの集合 D_i**

ある時点 t_i におけるデフォルトの集合 D_i とは，リテラル $l \in D$ の集合であり，時点 t_i において投機的計算に使用しているリテラルであり，次式により求める．

$$D_i = T_{R_i} - R_i$$

集合 D_i の要素それぞれが正しい確率を $q(t_i)$ とすると，時点 t_i においてデフォルトの回答が正しい確率 $p(t_i)$ を

$$p(t_i) = \prod_{d \in D_i} q(t_i)$$

で与える．

次に，時点 t_i における意思決定を行うためのくじを定義する．

[定義 3.2] **意思決定くじ l**

時点 t_i における意思決定のためのくじ l_{t_i} を，次の 3 つ組 $l_{t_i} =< X, P_i, E(l_{t_i}) >$ で定義する．

- X は結果の組でありその要素は“コミットする”か“コミットしない”かである．また，コミットするときに得られる効用を $x(t_i)$，コミットしないときに得られる効用を $y(t_i)$ とする．
- P_i は確率の組でありその要素は“コミットする”確率 $p(t_i)$，“コミットしない”確率 $1 - p(t_i)$ である．
- $E(l_{t_i})$ はくじの期待効用であり次式で与えられる．

$$E(l_{t_i}) = p(t_i) \cdot u(x(t_i)) + (1 - p(t_i)) \cdot u(y(t_i))$$

[定義 3.3] **意思決定集合 Ω**

いま時点を t_s とし，エージェントが与えられた問題を解決しなければならない締切 (deadline) を t_d とすると，意思決定くじ l の集合：

$$\Omega_s = \{l_{t_s}, \cdots, l_{t_d}\}, 1 \leqq s \leqq d$$

を時点 t_s における**意思決定集合**と呼ぶ．

このとき，エージェントは，期待効用仮説に基づき“くじ” l の期待値 $E(l)$ が最大となるくじ l_{t_b} を選択する．すなわち，時点 t_b において意思決定する．この時点 $t_b (t_s \leqq t_b \leqq t_d)$ を時点 t_s における**コミットタイム**と呼び次式を満たす．

$$t_b = \arg\max_{t_i} \{E(l_{t_s}), \cdots, E(l_{t_i}), \cdots, E(l_{t_d})\}$$

3.2.2 実時間意思決定

意思決定集合 Ω_s は，時点 t_s において既に得られている回

答を用いて意思決定くじを作成し集めたものである．したがって，この集合 Ω_s より得られるコミットタイム t_b は時点 t_s におけるコミットするのに最適な時点である．しかし，時点が進むにつれて回答が得られる場合があるため，このとき得られたコミットタイム t_b が常に最適な時点であるとは言えない．そこで，常に最適なコミットタイムを求めるために意思決定集合を更新する．

意思決定集合 Ω の更新アルゴリズム

Step1. エージェントは他のエージェントに質問を送ると同時に意思決定集合 Ω を求め，コミットタイム t_b を算出する．

Step2. 現在の時点を t_c としたとき，t_b が

- $t_b \leqq t_c$ ならば，ただちにコミットする (意思決定プロセス終了)
- $t_b > t_c$ ならば，コミットせず時点 t_b まで待機する．

Step3-1. 時点 t_b までに返答が得られなかった場合，時点 t_b になったところでコミットする (意思決定プロセス終了)

Step3-2. 時点 t_b までに返答が得られた場合，改めて意思決定集合 Ω を求め，コミットタイム t_b を算出し直し Step2. へ戻る．

3.2.3 時間 t を変数とする関数の導入

確率 P や結果の集合 X は時間によって変わるので，時間 t を変数とする関数を導入する．時点 t_i において，D_i の要素それぞれが正しい確率を $q(t)$ で与えると，コミットする確率 $p(t)$ を次式のように定義する．

$$p(t) = \prod_{d \in D_i} q(t)$$

また，くじ "l" の期待値の時間関数 $E(t)$ は，

$$E(t) = \int_{t_s}^{t_d} \{p(t) \cdot u(x(t)) + (1 - p(t)) \cdot u(y(t))\} dt$$

となり，コミットする最適な時間 t_b は $E(t)$ が最大となるときの t の値である．

期待値の関数 $E(t)$ はコミットタイムを求めるための関数であるので，この関数を**コミットメント関数**と呼ぶことにする．ここで得られるコミットタイム t_b は，他のエージェントから回答が得られる度にコミットする確率 $p(t)$ が変化するため時点 t_i における最適な時点であると考えられる．したがって，意思決定集合 Ω と同様にコミットメント関数 $E(t)$ の更新が必要となるが，その更新アルゴリズムはほぼ同様に書くことができる．

3.3 実時間意思決定問題

ここでは，コミットメント関数を具体的に定める．まず，通常型と先行投機型の比較を行うために時点 t_i での意思決定問題の定式化を行う．時点 t_i における意思決定問題とは，ホスト・エージェントがコミットするかコミットしないかを決定する問題であり，メンバ・エージェントからの回答を必要とするホスト側の意思決定問題である．ここで，ホストとは問題を与えられたエージェントであり，メンバはホストが質問を行う対象となるエージェントのことを指す．

本システムを Java 言語により実装した構成図を図 1 に示す．GUI はユーザとのインタフェースであり，通信にはチャットを用いている．また，SOL 導出には Java 言語で書かれた

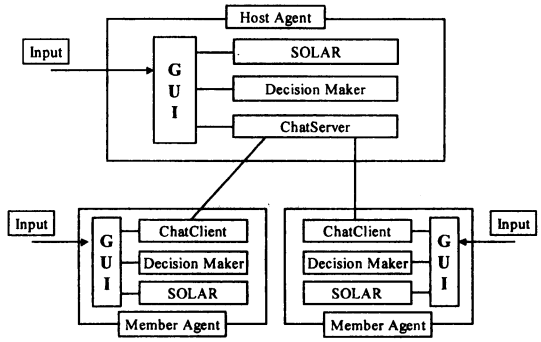

図 1 システム構成図

SOLAR [3] を利用している．

ホストは問題解決を行う (プランを実行する) と**報酬**を得られ，メンバはホストからの質問に対し回答すると**報酬**が得られるとする．しかし，メンバは他のエージェントからも質問を受けていたり，他の問題を与えられていたりするものとする．そして，どの問題から処理をするかは，得られる報酬によってメンバが独自に判断するものとする．すわなち，自らの効用が最大となるように行動するものとする．本研究では．これをエージェントの**合理性**と言う．また，プランの実行は時間の経過とともに難しくなるものとし，プランのキャンセルには**ペナルティ**が発生するものとする．

ここで，問題解決が成功するとは，全てのメンバからの回答が得られ，それによって SOL 導出により得られたプランが実行できることである．

a) [定式化パラメータ]

時点 $t(t_s \leqq t \leqq t_d)$ での意思決定問題を表すパラメータは以下のようになる．

$a(t)$: プランが成功したときにホストが得られる報酬

$b(t)$: ホストに回答したときにメンバが得られる報酬

$c(t)$: プランをキャンセルする場合に発生するペナルティ

$d(t)$: プランが達成できないことに対するペナルティ

$p(t)$: ホストがプランをコミットする確率

$r(t)$: プランが成功する確率

$s(t)$: 回答が揃う確率

意思決定問題には，ホストがコミットするタイミングの違いにより通常型と先行投機型の 2 つのタイプが存在する．通常型では，ホストはメンバに質問を行ったあと，メンバからの回答が届いてから SOL 導出によってプランを導出し実行する．しかし，メンバからの回答には何らかの理由により遅延が生じるため，例え締め切り時間内に全ての回答を得られたとしても，ホストがプランを実行できる確率は $q(t)$ なので確実に問題解決が成功するとは限らない．先行投機型では，ホストはメンバに質問を行ったあと，メンバからの回答を待たずにデフォルトの回答を用いて直ちに SOL 導出によってプランを求め実行する．デフォルトの回答が全て正しい場合には，問題解決は成功とい

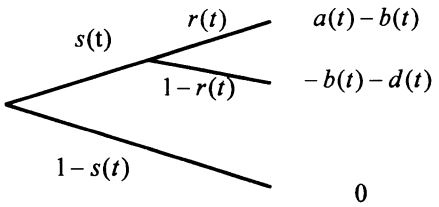

図 2　通常型の期待利得

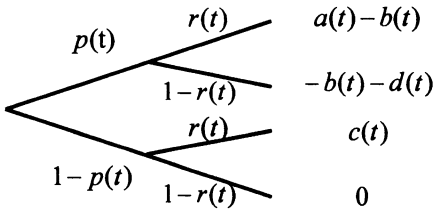

図 3　先行投機型の期待利得

うことになるが，デフォルトの回答が一部もしくは全部が異なり，そのためにプランが変更されれば問題解決は失敗となるためプランを取り消してペナルティ$c(t)$を支払わなければならない．また，どちらの場合でもプランを実行できなければペナルティ$d(t)$が発生する．

b)　[通常型と先行投機型の期待利得]

図 2,3 は意思決定問題においてホストが得られる利得をツリー状に場合分けしたものである．

図 2 は通常型の場合にホスト得られる利得であり，締め切り時間内に回答が全部揃い，かつ，プランが実行可能な場合にのみ報酬が得られ，実行できなければメンバに対し報酬を支払うだけになる．通常型における期待利得は次の通りである．ここで，t_sは問題が与えられた時点，t_dは締め切りの時点とする．

$$E_N(t) = \int_{t_s}^{t_d} [s(t)\{r(t)(a(t)+d(t)) - b(t) - d(t)\}]\,dt$$

図 3 は先行投機型の場合にホストが得られる利得であり，時点tにおけるデフォルトの回答が全て正しく，かつ，プランが実行可能な場合のみ報酬が得られる．また，時点tにおけるデフォルトの回答が間違っていてプランが実行可能な場合にはペナルティが発生するが，本研究ではデフォルトの回答が間違っている場合にはコミットしないため，コミットした場合に生じるペナルティを防ぐことができたと考える．先行投機型における期待利得は次の通りである．

$$E_{SC}(t) = \int_{t_s}^{t_d} [p(t)\{r(t)(a(t)+d(t)) - b(t) - d(t)\} + (1-p(t))r(t)c(t)]dt$$

本研究では，この期待利得が最大となる時点において意思を決定するものとしている．従って，コミットする時点の条件は，

$$\begin{cases} E_{SC}(t) \geq E_N(t), \\ t = \arg\max_t E_{SC}(t) \end{cases}$$

で与えられる．

表 1　各リテラルの意味

$travel(a)$	a が旅行する
$reserve(r,a)$	r が a のために部屋を予約する
$plan(twin\_room,[a,b])$	a,b のためにツインルームを予約する
$plan(single\_room,[a])$	a のためにシングルルームを予約する
$plan(no\_room,[\,])$	部屋を予約しない

次に，提案手法を用いることによりコミットタイムを求める例を示す．

3.4 例　　題

提案手法を適応する例題としてホテル予約問題 [5] を用いる．繁忙期におけるホテルの予約は時間の経過に伴い部屋が埋まっていくため，予約が難しくなる．したがって，本手法による効果を確認するのに適切な例題であると考えられる．ここでは，エージェントrの意思決定問題について取り上げコミットタイムを算出する．

ホテル予約問題の概要は次に示す通りである．

[例 3.1]　ホテル予約問題

- 3人のエージェントr,a,bが存在する．
- rはa,bに旅行するかどうかを尋ね，旅行するならばそれぞれのために部屋を予約する．さらに両方ともが旅行するならばツインルームを予約し，どちらか一方だけしか旅行しないのであればシングルルームを予約し，どちらも旅行しないのであれば部屋を予約しない．
- a,bはrに部屋を予約するかどうかを尋ね，rがツインであれシングルであれ，自分のための部屋を予約するならば旅行し，予約しないのであれば旅行しない．

ホテル予約問題におけるエージェントrの枠組は以下の通りである．また，それぞれのリテラルの意味は表 1 の通りである．

[エージェントrの枠組]

- $\Delta_D = \{travel(a), travel(b)\}, \Delta_U = \{\phi\}$.
- $D = \{travel(a), travel(b)\}$.
- Pは以下の節集合である．

$\neg travel(a) \vee reserve(r,a)$,

$travel(a) \vee \neg reserve(r,a)$,

$\neg travel(b) \vee reserve(r,b)$,

$travel(b) \vee \neg reserve(r,b)$,

$\neg reserve(r,a) \vee \neg reserve(r,b) \vee plan(twin\_room,[a,b])$,

$\neg reserve(r,a) \vee reserve(r,b) \vee plan(single\_room,[a])$,

$reserve(r,a) \vee \neg reserve(r,b) \vee plan(single\_room,[b])$,

$reserve(r,a) \vee reserve(r,b) \vee plan(no\_room,[])$.

このとき，エージェントrは，エージェントaもエージェントbも共に旅行すると信じており，ツインルームを予約するというプランを導出している．また，ツインルームを予約するときの報酬やデフォルトの回答が正しい確率を次のように定めた．

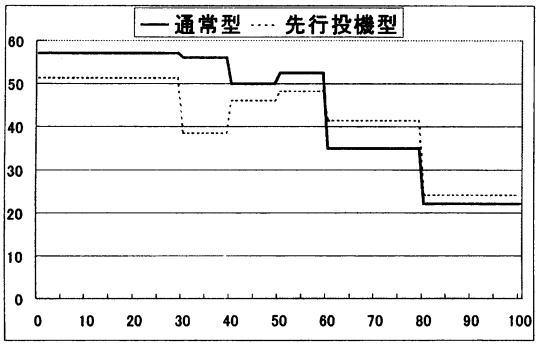

図 4　期待利得の時間推移

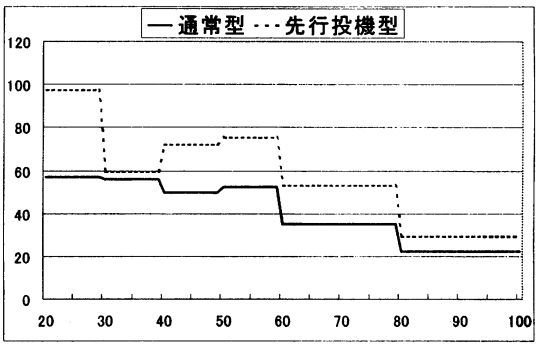

図 5　回答受信後における期待利得の時間推移

$$a(t)=\begin{cases}200 & (0\leqq t\leqq 60)\\ 150 & (60<t\leqq 100)\end{cases}\quad b(t)=\begin{cases}10 & (0\leqq t\leqq 50)\\ 5 & (50<t\leqq 100)\end{cases}$$

$$c(t)=\begin{cases}5 & (0\leqq t\leqq 30)\\ 10 & (30<t\leqq 60)\\ 20 & (60<t\leqq 100)\end{cases}\quad d(t)=\begin{cases}60 & (0\leqq t\leqq 40)\\ 100 & (40<t\leqq 80)\\ 120 & (80<t\leqq 100)\end{cases}$$

$$q_1(t)=\begin{cases}0.5 & (0\leqq t\leqq 30)\\ 0.6 & (30<t\leqq 60)\\ 0.7 & (60<t\leqq 100)\end{cases}\quad q_2(t)=\begin{cases}0.5 & (0\leqq t\leqq 40)\\ 0.7 & (40<t\leqq 100)\end{cases}$$

$$r(t)=\begin{cases}1.0 & (0\leqq t\leqq 30)\\ 0.7 & (30<t\leqq 80)\\ 0.6 & (80<t\leqq 100)\end{cases}\quad s(t)=\begin{cases}0.3 & (0\leqq t\leqq 30)\\ 0.5 & (30<t\leqq 80)\\ 0.6 & (80<t\leqq 100)\end{cases}$$

ここで，$q_1(t)$ は $travel(a)$ が真である確率を，$q_2(t)$ は $travel(b)$ が真である確率を表す．よって，コミットする確率は

$$p(t)=q_1(t)\cdot q_2(t)$$

となる．

この例におけるコミットタイムは，図 4 より $60<t<80$ である．従って，$60<t<80$ のどの時点でも意志を決定しても良い．このように期待利得の値が等しくなる時点が複数存在した場合は，より先行的に行動させたい場合はなるべく小さい時点を選択すればよいが，通常は，コミットする確率，すなわちデフォルトが正しい確率を上げるためにより大きな時点を選択するのが望ましいと考えられる．

通常型と先行投機型の期待利得の推移について考察する．この例ではエージェント a,b からの回答がまだ行われていない場合である．図 4 より，時点が小さいところでは通常型の方が先行投機型よりも期待利得が高くなっている．これは，デフォルトの回答のみを用いているためコミットする確率が低いためと考えられ，問題が与えられた時点でコミットする場合に比べキャンセル料の発生を抑えることが出来る．

次に，時点 $t=20$ において，エージェント a から $free(a)$ が真であると回答を得たとする．このとき，更新アルゴリズムにより意思決定集合の更新が為される．つまり，コミットする確率が $p(t)=q_2(t)$ となるため期待利得の時間推移は図 5 のように変化する．図 5 より，コミットタイムは $20<t<30$ となる．この更新によりコミットタイムが繰り上げられ期待利得が上昇しているのが図 4,5 を比較することにより見て取れる．これは，エージェントは自らの効用を最大にするように行動するというエージェントの合理性に適うものである．

4. ま と め

投機的計算を用いたマルチエージェントシステムにおけるこれまで研究では，投機的計算と SOL 導出によって得られたプランを実行するタイミングについてはあまり議論されてこなかった．そこで，本論文ではエージェントに対し期待効用理論に基づき効用を与えることにより，実時間で最適なタイミングを算出する手法を提案した．また，他のエージェントからの回答が得られる毎に期待効用の関数を更新することにより，常に最適な時点で意思決定を行うことができるアルゴリズムを提案した．さらに，例題としてホテル予約問題を取り上げて本手法を適応し最適なタイミングを求めた．

本論文では，ユーザが与える確率は主観確率であり更新されることはない．そこで，ユーザの行動履歴に基づいた主観確率の更新について今後議論していく必要がある．

文　献

[1] Inoue, K.: Linear resolution for consequence finding, *Artificial Intelligence*, 56, pp.301–353, 1992.

[2] Inoue, K., and Iwanuma, K.: Speculative Computation through Consequence-finding in Multi-agent Environments, *Annals of Mathematics and Artificial Intelligence*, 42(1-3), pp.255–291, 2004.

[3] Nabeshima, H., Iwanuma, K., and Inoue, K.,: SOLAR : A Consequence Finding System for Advanced Reasoning, *Proceedings of the 11th International Conference (TABLEAUX 2003), Lecture Notes in Artificial Intelligence*, 2796, pp. 257–263 (2003).

[4] Satoh, K., Inoue, K., Iwanuma, K., and Sakama, C.: Speculative computation by abduction under incomplete communication environments , *Proceedings of the Fourth International Conference on MultiAgent Systems (ICMAS 2000)*, pp.263-270, IEEE Computer Society, 2000.

[5] Satoh, K. and Yamamoto, K.: Speculative Computation with Multi-Agent Belief Revision, *Proceeings of AAMAS 2002*, pp.897–904 (2002).

[6] 岡田 章 : ゲーム理論, 有斐閣, 1996.

[7] 村尾 拓哉, 北村 泰彦, 東尾 潔, 辰己 昭治 : 先行投機的行動におけるリスク管理手法 : 会議室予約問題における事例研究, 合同エージェントワークショップ&シンポジウム *2003 (JAWS2003)*, pp.245–251, 2003.