人工知能学会誌 Vol. 11 No. 4 (July 1996) 〔別刷〕

論 文

Technical Papers

動的環境における成功確率を用いた熟考の制御

Controlling Deliberation with the Success Probability in a Dynamic Environment

山田 誠二*
Seiji Yamada

* 東京工業大学大学院総合理工学研究科知能システム科学専攻
Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology, Yokohama 226, Japan.

1995年8月23日 受理

Keywords: reactive planning, exectution, the success probability.

Summary

This paper describes a novel method *SIP* to interleave planning with execution in a dynamic environment. To determine the timing of interleaving them, we uses the success probability, *SP*, that it successfully executes a plan in an environment. *SP* is formally defined with likelihood that all operators in a plan are executable in an environment, and we develop a method to compute it inexpensively. An interleave planning system integrates reactivity with deliberation depending on dynamics of an environment. We require planning for intelligent behavior, and need to integrate reactivity with deliberation. Unfortunately, few solution have been proposed to this problem. Our approach gives a solution by interleaving planning with execution. We assign input probabilities to effects of actions and persistence of objects in an environment. Plans are transformed into Bayesian networks on which their *SP*s are computed in $O(n)$ time: n is plan size. A system switches planning to execution when *SP* falls below an execution threshold. After the execution, a system observes an environment, and starts planning again.

1. はじめに

AIとロボティクスにおいて，プランを極力作らずに行動(behavior)，あるいはリアクティブルールと呼ばれる「状況→行為」のルールを適用・実行することにより，迅速な挙動を実現するリアクティブプラニングの研究が活発である[Agre 87, Brooks 86, 山田 93]．しかし，このようなアプローチでは，予測を含む高度な知能の実現は困難であり，古典的プラニングとリアクティブプラニングの統合，すなわち熟考(deliberation)と即応(reactivity)の統合を図ることが重要である．本論文では，その統合の一手法として，目標スケジューリングをプラニングとする動的環境において，プランの成功確率を用いてプラニングと実行をインタリーブ(interleave)する手法 *SIP* (Success probability based Interleave Planning)を提案する[山田 91, Yamada 96]．このようなインタリーブは，プラニング[McDermott 78]や実時間探索[Korf 90]の分野において提案されてきたが，重要な問題である熟考の制御，つまりいつプラニングと実行を切り換えるかに関する研究例はほとんどない．本研究では，プランの実行が成功する確率がしきい値を下回ったときに，プラニングを実行に切り換える手法を示す．このプランの成功確率は，プランをベイジアンネットワークに変換し，そのうえでプラン長の線形オーダで計算可能であり，さらに環境の変化速度を考慮した計算により，環境の変化に適応した熟考の制御が可能となる．

2. 対象領域

リアルタイム(知識ベース)システムの対象領域[Laffy 88, 横田 94]およびエージェント設計において一般的な実験環境である(単純)タイルワールド[Kinny 91, Pollack 90]をもとに，*SIP*の対象領域である環境を次のように設定する．ここでは，自分で

外界を観測し，行為の決定・実行を行うシステムを**エージェント**と呼び，エージェントを取り巻く外界を**環境**と呼ぶ．

【定義1】 動的環境とエージェントの目的 非同期に出現しては消滅する複数の目標 G_i が存在する問題空間を**動的環境**とする．各目標 G_i はそれぞれ非負の**価値** V_i を持っており，エージェントは，目標が消滅するまでにそれを処理することにより，その価値を獲得する．また，**エージェントの目的**は，単位時間当りに獲得する価値を高くすることである． □

次に，*SIP*の**オペレータ，プラン，プラニング**を定義する．なお以降では，命題 P に対し，$+P$ は P が真，$-P$ は P が偽であること，$\neg P$ は P の否定を意味し，さらに $*P \in \{P, \neg P\}$ とする．

【定義2】 プラニング ある時点で観測された複数の目標に対し，できる限り高い価値を獲得できるような目標の処理順序を探すこと． □

【定義3】 目標オペレータとプラン *SIP*では，**目標オペレータ**と呼ばれる1種類のオペレータのみを用いる．目標オペレータ $O_i=(C_i, D_i, A_i)$ は，一つの目標の処理に対応するSTRIPS-likeなオペレータで，リテラルのリストである**条件リスト** C_i，**削除リスト** D_i，**追加リスト** A_i からなる．各リストの意味は，STRIPS [Fikes 71]に準拠する．条件リストのみ，負のリテラルを含んでよい．

また，追加リスト中に価値の獲得を意味する**獲得リテラル** s を導入する．よって，s 以外のリテラルは，価値獲得とは直接関係ない副作用を表す．さらに，目標オペレータ O には，処理時間を返す関数である**処理時間関数** $et(O)$ が割り当てられている．

具体化された目標オペレータの系列を**プラン**とする．プラン中の O_i を構成する C_i, D_i, A_i 中のリテラルを，それぞれ**条件，削除，追加リテラル**と呼び，$*L_{C_i}$, L_{D_i}, L_{A_i} と表す． □

上記のような設定において，**熟考**とは，目標を処理する最適な順序を考えることを，**即応**とは，処理順序を検討せずに即座に実行を開始することを意味する．

3. 成功確率を用いたインタリーブプラニング：*SIP*

3・1 *SIP* 手続き

SIP により行動を決定する *SIP* エージェントの構成を図1に示す．**実行モジュール**はプランナが生成したプランを実行し，**観測モジュール**は他のモジュールとは並列に外界を常時観測しており，センサ情報を

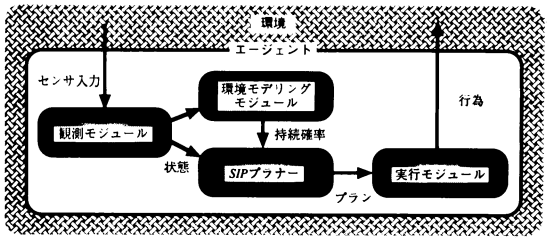

図1 *SIP* エージェント

一定時間ごとに内部表現で記述された状態に変換する．そして，プランナおよび環境モデリングモジュールの要請により，最新の状態を出力する．**環境モデリングモジュール**では，観測モジュールから得られる環境の情報に基づいて，環境の構造を獲得するが，*SIP*では後述する持続確率の推定のみを行っている．具体的な方法は，別稿[山田 96]に譲る．獲得された持続確率は，定期的に *SIP* プランナに渡される．*SIP*プランナは，*SIP*によるプラニングを行う．

動的な世界において，我々はいつプランを実行に移すのだろうか．この問に対し，「我々は今考えている最適なプランの実行の成功する確率が高いうちは，まだプラニングを続け，それがある程度まで下がったときに実行を開始する」という基準を採用する．よって，*SIP* はプランの成功確率によりプラニングと実行の切換えタイミングを決定する．まず，全体の手続きを図2に示す．図中の *Observation* で，*SIP* プランナは観測モジュールから内部表現を受け取り，*Getting environment structure* で，環境モデリングモジュールから持続確率を受け取る．

探索戦略は，幅 w の前向き横型ビーム探索である．前向き探索により，部分プランの初期状態への適用が可能となる．観測された環境を初期状態として，プラニングを進めていくが，1ステップの展開ごとに，展

```
A procedure SIP(G, w, τ)
  (G: a goal state, w: a beam width, τ: an execution threshold)
  while true do
    begin
      Observation;
      Getting environment structure;
      CS ← {(the observed state, [ ])}; % CS is a set of sp-pair: (State, Plan).
      P ← [ ];
      MSP ← 2;
      while MSP > τ ∧ P is not a complete plan in CS do
        begin
          NS ← all sp-pairs expanded from CS with operator applications;
          Computing SP and expected values for NS;
          CS ← w sp-pairs with high expected values in NS;
          MSP ← the success probability of the plan P with the maximum
                expected value in CS
        end
      [O1,...,On] ← P;
      i ← 1;
      while i ≠ n+1 ∧ the literal s of Oi-1 is achieved do
        begin
          executing Oi;
          i ← i +1
        end
    end
```

図2 *SIP* 手続き

開されたプランそれぞれにその**成功確率**と**期待価値**(expected value)が算出され，期待価値の上位 w 個のプランが選択されて展開が繰り返される．プランの成功確率と期待価値は，**次節**で説明する．各レベルの展開において，期待価値最大のプランの成功確率が実行のしきい値 $\tau \in [0, 1]$ 以下になるか，またはそのプランが完全プランである場合，プラニングを中止して，その期待価値最大のプランを実行する．プランの実行後，再び観測モジュールから状態を受け取り，初期状態が更新され，プラニングが繰り返される．ここで，**完全プラン**とは，観測された目標をすべてたどるプランであり，**部分プラン**とは，部分的にたどるプランである．

3・2 プランの成功確率

まず，目標オペレータ実行の成功を定義する．

【定義 4】 目標オペレータ実行の成功 目標オペレータ O_i の実行後，その追加リスト中の獲得リテラル s_i が成り立つことを O_i の**実行の成功** S_i とする． □

次に，プラン実行の成功とその成功確率は次のように定義される．以降で $Pr(A \wedge B) = Pr(A, B)$ である．

【定義 5】 プラン実行の成功とプランの成功確率 プラン $[O_1, \cdots, O_n]$ を実行した結果，すべての O_i $(1 \leq i \leq n)$ の成功 S_i が真であること，つまり $+S_1 \wedge \cdots \wedge +S_n$ を**プラン実行の成功**という．また，その確率 $SP = Pr(+S_1, \cdots, +S_n)$ を**プランの成功確率**とする． □

図 2 に示したように実行モジュールはプラン中の目標オペレータを一つずつ実行してその成功を確認する．そして，プラン中のすべてのオペレータの実行が成功するか，あるいは途中で一つでもオペレータの実行が失敗したときに，プラン実行を終了する．ただし，目標オペレータは，目標を処理する順序を示すだけであり，目標の具体的な処理手続きは，実行モジュールに記述されているとする．

上記の実行手続きと後述する **4•1 節の仮定 3** により，プランの実行によって得られる期待価値は，次のようになる．

【定義 6】 プランの期待価値 プラン $[O_1, \cdots, O_n]$ 中の O_i の実行により得られる目標 G_i の価値が V_i のとき，**プランの期待価値** $E[V]$ は次のようになる．

$$E[V] = Pr(+S_1) \cdot V_1 + Pr(+S_1, +S_2) \cdot V_2 + \cdots + Pr(+S_1, \cdots, +S_n) \cdot V_n \quad \square$$

4. ベイジアンネットワークによるプランの表現と成功確率の計算

本章では，熟考制御の要となるプランベイジアンネットワークとそのうえでの成功確率の計算方法を説明する．

4・1 プランベイジアンネットワーク

まず，基本概念の定義を示す．

【定義 7】 時制命題 環境において正のリテラル L が，時間 t で成り立つまたは成り立たないという命題を**時制命題** $\langle L, t \rangle$ または $\langle \neg L, t \rangle$ とする． □

【定義 8】 因果関係と時間 プラン中の条件リテラル L_{C_j} が，O_i により追加された L_{A_i} である場合，または $\neg L_{C_j}$ の L_{C_j} が O_i により削除された L_{D_i} である場合，**因果関係** $L_{A_i} < L_{C_j}$，または $L_{D_i} < \neg L_{C_j}$ $(i < j)$ があるとする．また，プラン $P = [O_1, \cdots, O_n]$ に対し，O_1 の**実行開始予定時間** t_0，O_i $(2 \leq i \leq n)$ の**実行終了予定時間** $t_i = t_0 + \sum_{k=1}^{i} et(O_k)$，そして観測可能な任意のリテラル L が環境において真または偽になったことが観測された**観測時間** $t(*L)$ の集合を P の**時点集合**と呼ぶ．$t(*L)$ は，プラニングおよび実行に並列に行われる観測から得られる． □

成功確率の計算には，以下の確率が入力として必要である．これらは，既与であるか，エージェントが環境の観測により推定する．

【定義 9】 効果確率 $E\text{-}Pr(O_i, L)$ 目標オペレータ O_i により追加(削除)されたリテラル L が，O_i の実行終了時に環境において真(偽)である確率であり，行為の信頼性を表す． □

【定義 10】 観測確率 $O\text{-}Pr(*L)$ 観測されたリテラル $*L$ が，観測時に環境において真であった確率．観測の精度を表す． □

【定義 11】 持続確率 $P\text{-}Pr(*L, T)$ ある時間に環境で真(偽)となったリテラル $*L$ が，そのエージェントのオペレータにより操作されずに時間 T たった後でも真(偽)である確率．環境の変化の程度を表す． □

以上のプラン，時点集合，入力確率が，プランベイジアンネットワーク生成および成功確率計算の入力である．ベイジアンネットワーク[Charniak 91, Pearl 88, Shapiro 91]とは，命題変数をノード，それらの直接的因果関係をアークとする非循環(acyclic)な有向グラフである．アークには，条件つき確率を割り当てることにより，因果関係の強さを表現する．ベイジアンネットワーク上で計算されるのは，真偽が既知である証拠ノードの集合 $\{e_1, \cdots, e_n\}$ が与えられたときの各ノード x の事後確率ベクトル $BEL(x) = (Pr(+x | e_1 \wedge \cdots \wedge e_n), Pr(-x | e_1 \wedge \cdots \wedge e_n))$ である．この計算は，証拠ノードからのデータの伝搬により行われる．

以下では，V はノードの集合，E はアークの集合，そしてノード v_1, v_2 について，$e(v_1, v_2) \in E$ は，有向アーク $v_1 \rightarrow v_2$ を表す．さらに，命題 $Ob(*L, t)$ は，リテラル L が時間 t に真または偽になったことが観測されたこと，命題 $Ex(O)$ はオペレータ O が環境で実行可能であることを意味する．

【定義 12】 プランベイジアンネットワーク PB

プラン $P=[O_1, \cdots, O_n]$ のプランベイジアンネットワーク $PB=(V, B)$ は，以下の定義 13, 14 の要素からなる有向非環境グラフである． □

なお，以下における時間 t_i, $t(L)$ は，定義 8 を参照．

【定義 13】 ノードとアーク

- **実行ノード**：$\langle Ex(O_i), t_{i-1}\rangle \in V$ $(1 \leq i \leq n)$ である．これを**実行ノード**という．入出次数ともに，1 以上．
- **条件ノード**：プラン中のすべての条件リテラル $*L_{C_i}$ $(1 \leq i \leq n)$ について，$\langle *L_{C_i}, t_{i-1}\rangle \in V$，かつ $e(\langle *L_{C_i}, t_{i-1}\rangle, \langle Ex(O_i), t_{i-1}\rangle) \in E$ である．$\langle *L_{C_i}, t_{i-1}\rangle$ は，**条件ノード**と呼ばれる．入出次数ともに，1．
- **追加ノード**：プラン中のすべての追加リテラル L_{A_i} $(1 \leq i \leq n)$ について，$\langle L_{A_i}, t_i\rangle$ かつ $e(\langle Ex(O_i), t_{i-1}\rangle, \langle L_A, t_i\rangle) \in E$ である．$\langle L_A, t_i\rangle$ を O_i の**追加ノード**という．入次数 $=1$．出次数は，任意．
- **削除ノード**：プラン中のすべての削除リテラル L_{D_i} $(1 \leq i \leq n)$ について，$\langle \neg L_{D_i}, t_i\rangle \in V$，かつ $e(\langle Ex(O_i), t_{i-1}\rangle, \langle \neg L_{D_i}, t_i\rangle) \in E$ である．$\langle \neg L_{D_i}, t_i\rangle$ を O_i の**削除ノード**という．入次数 $=1$，出次数は任意．
- **観測ノード**：条件ノード $\langle *L, t_{i-1}\rangle$ が成り立つことが観測によりわかった場合，$\langle Ob(*L), t(*L)\rangle \in V$ かつ $e(\langle Ob(*L), t(*L)\rangle, \langle *L, t_{i-1}\rangle) \in E$ である．$\langle Ob(*L), t(*L)\rangle$ は，出次数 1 の根ノードで，**観測ノード**と呼ばれる．
- **因果アーク**：$L_{A_i} < L_{C_j}$ のとき $(i<j)$, $e(\langle L_{A_i}, t_i\rangle, \langle L_{C_j}, t_{j-1}\rangle) \in E$ であり，また $L_{D_i} < \neg L_{C_j}$ のとき $(i<j)$, $e(\langle \neg L_{D_i}, t_i\rangle, \langle \neg L_{C_j}, t_{j-1}\rangle) \in E$ である． □

以降で，$e(x, y)$ に割り当てられる条件つき確率を行列

$$M_{y|x}=\begin{pmatrix} Pr(+y|+x) & Pr(-y|+x) \\ Pr(+y|-x) & Pr(-y|-x) \end{pmatrix}$$

で表す．

【定義 14】 ノードとアークの確率 以下のように，ノードとアークに，確率が割り当てられる．

- **観測確率**：観測ノード $\langle Ob(*L), t(*L)\rangle$ に，$BEL(\langle Ob(*L), t(*L)\rangle)=(O\text{-}Pr(*L), 1-O\text{-}Pr(*L))$ を設定．
- **効果確率**：$x=\langle Ex(O_i), t_{i-1}\rangle$ とその子であるリテラル L の追加または削除ノード y 間のアークに割り当てられる条件つき確率は，

$$M_{y|x}=\begin{pmatrix} E\text{-}Pr(O_i, *L) & 1-E\text{-}Pr(O_i, *L) \\ 0 & 1 \end{pmatrix}$$

である．

- **持続確率**：$x=\langle L_{A_i}, T_1\rangle$（または，$\langle Ob(L), T_1\rangle$）と $y=\langle L_{C_j}, T_2\rangle$ 間，あるいは $x=\langle \neg L_{D_i}, T_1\rangle$（または，$\langle Ob(\neg L), T_1\rangle$）と $y=\langle \neg L_{C_j}, T_2\rangle$ 間のアークに割り当てられる条件つき確率は，

$$M_{y|x}=\begin{pmatrix} P\text{-}Pr(*L, T_2-T_1) & \\ 0 & \\ & 1-P\text{-}Pr(*L, T_2-T_1) \\ & 1 \end{pmatrix}$$

である．

- **条件つき確率**：実行ノード Ex が，m 個の条件ノード c_i $(1 \leq i \leq m)$ を持つとき，それらの間に，条件つき確率の分布

$$Pr(+Ex|c_1, \cdots, c_m)=\begin{cases} 1 & \text{if} + c_1 \wedge \cdots \wedge + c_m \\ 0 & \text{otherwise} \end{cases},$$

$$Pr(-Ex|c_1, \cdots, c_m)=\begin{cases} 0 & \text{if} + c_1 \wedge \cdots \wedge + c_m \\ 1 & \text{otherwise} \end{cases}$$

が割り当てられる． □

以上の定義により，プランから一意にプランベイジアンネットワークが構成できる．また，上記の条件つき確率は，以下の仮定を用いている．

仮定 1：観測ノード間，同一オペレータの条件，追加ノード間には，直接的依存関係はない．

仮定 2：オペレータの実行は，その条件がすべて真のとき，かつそのときに限り可能である．

仮定 3：エージェントは，自分の行為が成功した世界のみを予測する．

厳密に確率的時制射影(probabilistic temporal projection)[Hanks 90]や確率プラニング(probabilistic planning)[Kushmerick 90]を行うと，多重世界が指数関数的に爆発してしまう[Hanks 90]が，本手法では**仮定 3** により，精度は犠牲にするものの大幅な枝刈りがなされる．また，確率プラニング[Kushmerick 90]のように時間とともに成功確率が増加することはない．

ベイジアンネットワーク生成の具体例を示す．

プラン

$$P=[O_1, O_2, O_3]$$
$$=[([a_{C_1}, b_{C_1}], [c_{D_1}], [d_{A_1}]),$$
$$([d_{C_2}, \neg e_{C_2}], [], [f_{A_2}]),$$
$$([d_{C_3}, f_{C_3}, \neg c_{C_3}], [], [g_{A_3}])]$$

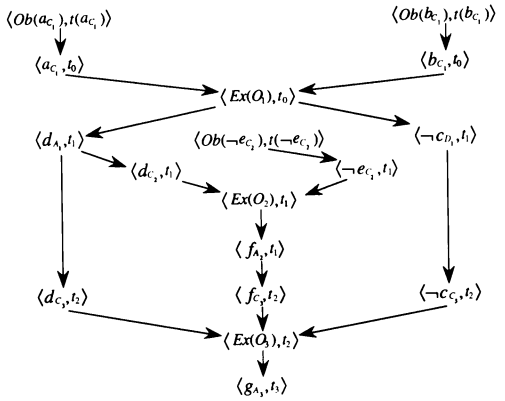

図 3　プランベイジアンネットワーク

因果関係

$$\{d_{A_1} < d_{C_3},\ c_{D_1} < \neg c_{C_3},\ f_{A_2} < f_{C_3}\}$$

と適当な時間点集合，入力確率から，図3のようなプランベイジアンネットワークが生成される．ただし，図中で条件つき確率は割愛している．

4・2　成功確率の計算

すでに我々は定義5において，プラン実行の成功を「プラン中のすべてのオペレータの実行が成功すること」と定義したが，成功確率はプランの実行時間に依存するので，ここで O_i の獲得リテラル s を s_i とし，さらに時間を明示して，S_i を$\langle s_i, t_i\rangle$と書き換え，プランの成功確率を「時間 t_0 でプランの実行を始めた場合に，プラン中のすべてのオペレータの実行が成功する確率」とする．よって，実行開始時間 t_0 のプラン $P=[O_1, \cdots, O_n]$ の成功確率は，$SP(P, t_0)=Pr(+\langle s_1, t_1\rangle, \cdots, +\langle s_n, t_n\rangle)$ となり，確率の連鎖規則により下式(1)が得られる．そして，さらに式(2)，式(3)が得られるが，これらの導出は，**付録**を参照．ただし，$\boldsymbol{e}_i=+\langle s_1, t_1\rangle \wedge \cdots \wedge +\langle s_i, t_i\rangle$とする．

$$\begin{aligned} SP(P, t_0) &= Pr(+\langle s_1, t_1\rangle, \cdots, +\langle s_n, t_n\rangle) \\ &= Pr(+\langle s_1, t_1\rangle) \cdot Pr(+\langle s_2, t_2\rangle | +\langle s_1, t_1\rangle) \\ &\quad \cdots\cdots Pr(+\langle s_n, t_n\rangle | +\langle s_1, t_1\rangle, \\ &\quad \cdots, +\langle s_{n-1}, t_{n-1}\rangle) \\ &= \prod_{i=1}^{n} Pr(+\langle s_i, t_i\rangle | \boldsymbol{e}_{i-1}) \qquad (1) \end{aligned}$$

$$\begin{aligned} &Pr(+\langle s_i, t_i\rangle | \boldsymbol{e}_{i-1}) \\ &= E\text{-}Pr(O_i, +\langle s_i, t_i\rangle) \\ &\quad \prod_{*L \in C_i} Pr(+\langle *L_{C_i}, t_{i-1}\rangle | \boldsymbol{e}_{i-1}) \qquad (2) \end{aligned}$$

$$\begin{aligned} &Pr(+\langle *L_{C_i}, t_{i-1}\rangle | \boldsymbol{e}_{i-1}) \\ &= \begin{cases} P\text{-}Pr(*L, t_{i-1}-t_h) & \text{(a)} \\ E\text{-}Pr(O_h, *L) \cdot P\text{-}Pr(*L, t_{i-1}-t_h) & \text{(b)} \\ O\text{-}Pr(*L) \cdot P\text{-}Pr(*L, t_{i-1}-t_h) & \text{(c)} \end{cases} \\ &\qquad (3) \end{aligned}$$

式(3)の(a)～(c)は，以下の場合である．ただし，$\langle *L_{C_i}, t_{i-1}\rangle$ の親ノードを N，その時点を t_h とする．

(a)：N が $\langle Ex(O_1), t_0\rangle \sim \langle Ex(O_{i-1}), t_{i-2}\rangle$ のいずれかの獲得リテラル s のノードであるか，または，$\langle Ex(O_1), t_0\rangle \sim \langle Ex(O_{i-1}), t_{i-2}\rangle$ のいずれかの条件ノードが $\langle *L_{C_i}, t_{i-1}\rangle$ の兄弟ノードである場合．

(b)：(a)を満たさず，かつ N が実行ノード $Ex(O_h, t_h)$ の追加ノードまたは削除ノードの場合．

(c)：(a)を満たさず，かつ N が観測ノードの場合．

式(1), (2), (3)より，成功確率 $SP(P, t_0)$ が計算できる．実行開始時間は，プラニングが進むにつれて変化するので，n ステップ展開時の成功確率の計算において，すでに算出されている $Pr(+\langle s_1, t_1\rangle, \cdots, +\langle s_{n-1}, t_{n-1}\rangle)$ も更新される．

また，***SIP***において，成功確率はインクリメンタルに計算可能で，その計算量は，プランの1ステップにつき，定数オーダである(**付録**参照)．よって，プラン長を n とすると，プラン全体の成功確率の計算量は $O(n)$ となる．これに対し，一般にベイジアンネットワーク上では，任意のノードの事後確率が計算可能であるが，無向閉路を含む場合，その計算量は，NP-hardになることが知られている[Cooper 90]．このように計算量的な意味でも，我々の成功確率の定義は良い性質を持っている．また，この計算量が線形オーダになるためには，**4・1節**の仮定が十分条件である．

4・3　検　　討

本節では，***SIP***の特徴について検討する．

成功確率の推移：局所最適なプランの成功が危うくなったときに実行を開始するというメカニズムが，***SIP*** により実現される．実行されるプランは，その時点でのビーム幅の範囲における期待価値最大という意味で局所最適なプランである．プランの成功確率の推移の例を図4に示す．横軸はプラン長(＝オペレータ数)，最大期待価値を持つプランの成功確率である．図において，例えば，実行のしきい値を0.8にすると長さ2のプランが，0.3にすると長さ6のプランが実行される．

環境の変化への適応：環境モデリングモジュールにより，目標の持続確率が絶えず更新される．よって，例えば，図5のような典型的な持続確率において，環境の変化が速くなると，実線から点線のように持続確率が変化する．これにより，成功確率の減少の傾きが，変化が速いときは急になり，遅いときは緩やかになる．つまり，実行のしきい値を固定しても，変化が速いときは短いプラン，遅いときは長いプランが生成される．

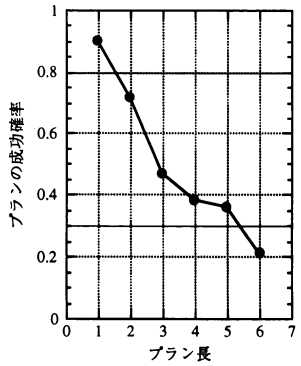

図 4 成功確率の推移

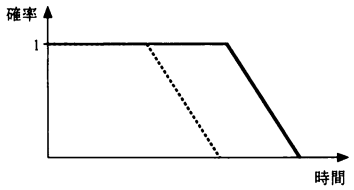

図 5 持続確率の変化

よって，最適な実行のしきい値をある一つの環境で見つけておけば，後は環境の変化の速さが変化しても，対応できる可能性がある．このことは，最適なしきい値が環境変化の速さの変化に対して頑健であることを意味し，別稿[山田 96]で実験的に検証される．

各処理の並列実行： *SIP*においてプランの実行中にプラニングを行うことは，その初期状態が決定困難であるため意味がない．また，プラニング中に現在立てているプランが実行不可能な場合，すぐに再プラニングを開始することにより，熟考を抑制することも考えられるが，*SIP*では環境のモデリングが正確であれば，このような事態を予測できるので必要ない．

最適しきい値の獲得： *SIP*では，最適な実行のしきい値の決定が問題であるが，これは領域依存と考えられ，経験的に調整するしかない．具体的には，実際のパフォーマンスを評価関数とした山登り法で局所最適値を探すことが考えられるが，かなりのコストがかかる．山登り法を用いる場合，パフォーマンスを返す実行のしきい値の関数が単峰性を持つことが望ましいが，この性質も別稿[山田 96]で単純タイルワールドにおいて実験的に示される．

5. 関 連 研 究

Drummond らの研究[Drummond 90]では，確率と効用により制御される時制射影により最適なプランを事前に生成し，そのプランから状況制御ルールが得られる．そこでは，本研究と類似した成功確率が定義されるが，非常に単純化されたオペレータを用いており，詳細な因果関係の記述は困難である．

三浦らは，視覚システムの不確実性と認識のコストを考慮したプラニングを提案している[三浦 92]．しかし，動的世界への拡張がされておらず，視覚以外のセンサに対する適用の可能性も不明である．それに対し，*SIP* は，動的世界に対応しており，センサモデルに依存しない一般性を持つ．

定数時間の探索結果をもとに移動しながら探索を行う実時間探索が，プラニングと実行をインタリーブするという意味で，*SIP*と関連がある．Korf の RTA* [Korf 90]は，定数の深さで先読みの探索を中止し，実行とのインタリーブを行う完全かつ正当(correct)なアルゴリズムである．しかし，探索の深さを，環境の変化に応じて変化させる手法は提案されていない．本研究は，成功確率を用いたプラニング制御の具体的手法を示している．また，石田は，移動する目標を追跡する完全な探索アルゴリズム MTS(Moving Target Search)を提案した[石田 93]．MTS では，不確実性により，推定距離が極小になる状態(推定凹部)が存在し，そこからの脱出の一手法として，目標の位置をコミットしてオフライン探索(＝熟考)を行うことを提案している．しかし，環境の変化に適応したオフライン探索の制御については述べていない．

Pollack ら[Pollack 90]によりタイルワールドで実験された IRMA アーキテクチャ[Bratman 88]は，次に処理すべき目標を決定するだけの単純な熟考だけを扱っており，熟考の制御は行っていない．さらに，Kinny ら[Kinny 91]は，一度決めた目標にどれだけ固執するかというコミットメントの程度と環境の変化の速さの関係に調べる実験をタイルワールドで行ったが，そこでも熟考は制御されていない．

Russell と Wefald の DTA*[Russell 91]は，決定理論に基づいて探索と移動のインタリーブを行う完全な探索アルゴリズムである．先読みが現時点の決定を覆す可能性を見積もり，その値がすべて負である場合には，探索を中止して実行を行い，そうでない場合には，見積り最大の枝を展開して探索を続ける．ただし，DTA* で見積もられるのは，現時点からの先読み探索の効用であり，現時点で生成されているプランの成功確率で制御する我々の手法とは大きく異なる．

確率プラニング[Kushmerick 90]は，初期状態と行為の結果に確率を導入し，目標を達成する成功確率が既与のしきい値よりも大きいプランを探索する．ただし，確率的な状態遷移を扱うため，問題空間が爆発し，計算効率は悪い．確率プラニングは，動的環境への対

応や熟考の制御を目的とはしていないが，任意時間性を持つため，一定時間で確率プラニングを走らせることにより，*SIP* における目標オペレータの効果確率の計算とパスプラニングとして利用できる．

6. ま と め

成功確率を用いて，古典的プラニングと実行を交互に行うインタリーブプラニング *SIP* を提案した．また，プランをベイジアンネットワークで表現し，プラン実行の成功確率を計算する方法について述べた．実世界では，エージェントが環境に関する完全な知識を持つことはあり得ない．よって，我々は行為に関する確率的な表現が必須であると考える．なお，*SIP* を用いたタイルワールドにおける実験が別稿[山田 96]で示される．

謝 辞

有益なコメントをいただいた礒田佳憲氏(NTT HI研究所)，平山勝敏先生(神戸商船大学)，椹木哲夫先生(京都大学)，淡誠一郎先生(近畿大学)，横田英俊氏(KDD研究所)，Decision-Theoretic Planning について教えていただいた原口 誠先生(北海道大学)，および MARK (Multi-Agent Research community in Kansai)で議論していただいた方々に感謝いたします．最後に，本研究を始める契機を与えて下さった辻三郎先生(和歌山大学)に深く感謝いたします．

◇ 参 考 文 献 ◇

[Agre 87] Agre, P. E. and Chapman, D.: Pengi: A Implementation of a Theory of Activity, *AAAI-87,* pp. 268-272 (1987).

[Bratman 88] Bratman, M., Israel, J. and Pollack, M.: Plans and Resource-Bounded Practical Reasoning, *Computational Intelligence 4,* pp. 349-355 (1988).

[Brooks 86] Brooks, R. A.: A Robust Layered Control System for a Mobile Robot, *IEEE Robotics and Automation,* Vol. 2, No. 1, pp. 14-23 (1986).

[Charniak 91] Charniak, E.: Bayesian Networks without Tears, *AI Magazine,* Vol. 12, No. 14, pp. 50-63 (1991).

[Cooper 90] Cooper, G. F.: The Computational Complexity of Probabilistic Inference Using Bayesian Belief Networks, *Artif. Intell.,* Vol. 42, pp. 393-405 (1990).

[Drummond 90] Drummond, M. and Bresina, J.: Anytime Synthetic Projection: Maximizing the Probability of Goal Satisfaction, *AAAI-90,* pp. 138-144 (1990).

[Fikes 71] Fikes, R. E. and Nilsson, N. J.: STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving, *Artif. Intell.,* Vol. 2, pp. 189-208 (1971).

[Hanks 90] Hanks, S.: Practical Temporal Projection, *AAAI-90,* pp. 158-163 (1990).

[石田 93] 石田 亨：移動目標探索アルゴリズムとその性能改善，人工知能学会誌，Vol. 8, No. 6, pp. 760-768 (1993).

[Kinny 91] Kinny, D. and Georgeff, M.: Commitment and Effectiveness of Situated Agents, *IJCAI-91,* pp. 82-88 (1991).

[Korf 90] Korf, R. E.: Real-Time Heuristic Search, *Artif. Intell.,* Vol. 42, pp. 189-211 (1990).

[Kushmerick 90] Kushmerick, N., Hanks, S. and Weld, D.: An Algorithm for Probabilistic Least-Commitment Planning, *AAAI-94,* pp. 1073-1078 (1994).

[Laffy 88] Laffy, T. J., *et al.*: Real-Time Knowledge-Based Systems, *AI Magazine,* Vol. 9, No. 1, pp. 27-45 (1988).

[McDermott 78] McDermott, D.: Planning and Action, *Cognitive Science,* Vol. 2, pp. 71-110 (1978).

[三浦 92] 三浦 純，白井良明：不確かさを考慮した視覚と行動のプラニング，人工知能学会誌，Vol. 7, No. 5, pp. 850-861 (1992).

[Pearl 88] Pearl, J.: *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference,* Morgan Kaufmann (1988).

[Pollack 90] Pollack, M. and Ringuette, M.: Introducing the TILEWORLD: Experimentally Evaluating Agent Architectures, *AAAI-90,* pp. 183-189 (1990).

[Russell 91] Russell, S. and Wefald, E.: *Do the Right Thing,* MIT Press (1991).

[Shapiro 91] Shapiro, S. C., ed.: 人工知能大辞典，pp. 1026-1034, 丸善(1987).

[山田 91] 山田誠二：インターリーブによるリアクティブ・プランニング，情処学会人工知能研究会，91-AI-79-8 (1991).

[山田 93] 山田誠二：リアクティブプラニング，人工知能学会誌，Vol. 8, No. 6, pp. 729-735 (1993).

[山田 96] 山田誠二，礒田佳徳，豊田順一：単純タイルワールドにおける *SIP* の実験的評価，人工知能学会誌，Vol. 11, No. 5 掲載予定 (1996).

[Yamada 96] Yamada, S.: Controlling Deliberation with the Success Probability in a Dynamic Environment, *AIPS-96,* to appear (1996).

[横田 94] 横田，橋本，浅見：リアルタイムプラニングにおける応答時間特性の解析手法，信学論(D-II)，Vol. J77-D-II, No. 4, pp. 838-849 (1994).

〔担当編集委員：諏訪 基，査読者：梅山伸二〕

◇ 付 録 ◇

式(2)の導出：ベイジアンネットワーク上の二つのノード n_i, n_j，そして真偽がわかっている証拠ノードの集合 Z について，n_i と n_j のノード間のすべてのパスが，次の(B1)，(B2)の妨害(block)ノードのいずれかを含む(妨害される)とき，n_i と n_j は互いに独立であるという性質，d 分割(d-separation)が知られている[Pearl 88, Shapiro 91].

(B1) それ自体とその子孫ノードが Z の要素でない収束ノード(複数の矢印の先端がぶつかるノード)．

(B2) 収束ノードではない Z の要素．

つまり，もしループ(無向閉路)上に妨害ノードがあれば，その

ループは分割され，もはやループではなくなる．以下では，各ノードの時点は省略されており，σ_i と Ex_i は，$\langle s_i, t_i\rangle$ と $\langle Ex(O_i), t_{i-1}\rangle$ の略である．

一般にベイジアンネットワーク上での任意の命題の確率計算には，妨害されないループの指数オーダの計算量が必要である[Pearl 88]．そこでまずプランベイジアンネットワーク PB 上での $Pr(+\sigma_i|\boldsymbol{e}_{i-1})$ の計算では，すべてのループは妨害されることを意味する次の定理を証明する．

[**定理 1**] プラン $[O_1, \cdots, O_n]$ の PB 上の証拠ノードの集合が $Z=\{+\sigma_1, \cdots, +\sigma_k\}$ $(1\leq k\leq n)$ のとき，PB 上のすべてのループは妨害ノードを含む．

《証明》 証拠ノード $Z=\{+\sigma_1, \cdots, +\sigma_i\}$ のとき，定義 14 より，Z のすべての祖先ノードは真となり，それらはすべて証拠ノードと考えてよい．PB は非循環であるから，任意のループは少なくとも一つの収束ノードを含む．定義 13 より，PB 上で入次数 2 以上の収束ノードになり得るのは実行ノードだけなので，その収束ノードのうち，最後(添字最大)のものを実行ノード Ex_m とする．このとき，Ex_m が Z の要素である場合とない場合の両方において，そのループが妨害ノードを含むことを示せば十分．

Ex_m が Z の要素の場合 $(1\leq m\leq k)$ は，ループ上にある，Ex_m の条件ノードが(B2)を満たし，妨害ノードとなる．また，定義 13 より，実行ノード Ex_q はその子孫に Ex_p $(1\leq p<q\leq n)$ を持たないことから，Ex_m が Z の要素でないとき $(k\leq m)$ は，Ex_m のすべての子孫は Z の要素ではない．よって，Ex_m 自身が(B1)を満たし，妨害ノードとなる．以上で，定理 1 が証明された． □

定理 1 より，証拠ノード $Z=\{+\sigma_1, \cdots, \sigma_{i-1}\}$ を用いて，$Pr(+\sigma_i|\boldsymbol{e}_{i-1})$ を計算するとき，PB は単結合(singly connected)と考えてよい．さらに，σ_i の子ノードと $\sigma_1, \cdots, \sigma_{i-1}$ との間のすべてのパスは，少なくとも一つの Ex_j $(i\leq j)$ により妨害されるので，子ノードから σ_i への伝搬はない．よって，σ_i の親ノードを通る伝搬だけを考えればよく，[Pearl 88]より，以下の制約式が成り立つ．

$$Pr(+\sigma_i|\boldsymbol{e}_{i-1}) = E\text{-}Pr(O_i, +\langle s_i, t_i\rangle) \sum_{c_1,\cdots,c_m} Pr(+Ex_i|c_1, \cdots, c_m) \prod_k Pr(c_k|\boldsymbol{e}_{i-1}) \quad \text{(i)}$$

ただし，c_k $(1\leq k\leq m)$ は，Ex_i の親ノード(=条件ノード)である．定義 14 の条件つき確率の定義より，$+c_1\wedge\cdots\wedge+c_m$ のときだけ $Pr(+Ex_i|c_1, \cdots, c_m)=1$ で，他の場合はすべて 0 であることと**仮定 1** より，式(i)が下のように展開され，式(2)が導かれる．

$$Pr(+\sigma_i|\boldsymbol{e}_{i-1}) = E\text{-}Pr(O_i, +\langle s_i, t_i\rangle) \prod_k Pr(+c_k|\boldsymbol{e}_{i-1})$$

(終わり)

式(3)の導出：次に，$Pr(+c_k|\boldsymbol{e}_{i-1})$ を求める．$Z=\{+\sigma_1, \cdots, +\sigma_i, +Ex_1, \cdots, Ex_i\}$ で考えると，Z により分割された部分ネットワークのうち，Ex_i とその親ノード c_k を含む部分ネットワークは，図 6 の 5 通りに分類できる．図中の(6-a)，(6-b)は，c_k の親ノードが追加ノード(または削除ノード) ad_h の場合で，(6-a)は ad_h が

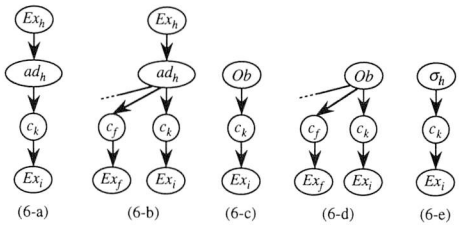

図 6 分割されたネットワーク

部分ネットワーク上の子孫として Z 中のノードを持たない場合であり，(6-b)は $Ex_f\in Z$ を持つ場合である．また，(6-c)，(6-d)は，c_k が観測ノードを親とする場合で，(6-a)，(6-b)と同様に，(6-c)はその観測ノードが，部分ネットワーク上の子孫として Z 中のノードを持たない場合，(6-d)は $Ex_f\in Z$ を持つ場合である．また，(5-e)は，c_k の親ノードが $\sigma_h\in Z$ の場合である．[Pearl 88]より，式(i)と同様に，下式が成り立つ．

$$Pr(+c_k|\boldsymbol{e}_{i-1}) = \begin{cases} \sum_{ad_h} Pr(+c_k|ad_h)\cdot Pr(ad_h|\boldsymbol{e}_{i-1}) \\ \sum_{\cup} Pr(+c_k|Ob)\cdot Pr(Ob|\boldsymbol{e}_{i-1}) \end{cases}$$

そして，定義 14 の持続確率の定義より，下式が得られる．ただし，c_k の親ノードの時点を t_h とする．

$$Pr(+c_k|\boldsymbol{e}_{i-1}) = \begin{cases} P\text{-}Pr(*L, t_{i-1}-t_h)\cdot Pr(+ad_h|\boldsymbol{e}_{i-1}) \\ P\text{-}Pr(*L, t_{i-1}-t_h)\cdot Pr(+Ob|\boldsymbol{e}_{i-1}) \end{cases}$$

よって，$Pr(+ad_h|\boldsymbol{e}_{i-1})$ と $Pr(+Ob|\boldsymbol{e}_{i-1})$ を求めればよい．図 6 の(6-b)と(6-d)の場合は，$Ex_f\in Z$ が真なので，定義 14 の条件つき確率より，そのすべての条件ノード，さらに ad_h, Ob もすべて真になる．よって，$Pr(+ad_h|\boldsymbol{e}_{i-1})=Pr(+Ob|\boldsymbol{e}_{i-1})=1$ となる．また，(6-e)では，明らかに $Pr(+\sigma_h|\boldsymbol{e}_{i-1})=1$ となる．次に，(6-a)の場合は，$Ex_h\in Z$ は真なので，定義 14 の効果確率より，$Pr(+ad_h|\boldsymbol{e}_{i-1})=E\text{-}Pr(O_h, *L)$ となる．最後に，(6-c)の場合は，定義 14 の観測確率より，$Pr(+Ob|\boldsymbol{e}_{i-1})=O\text{-}Pr(*L)$ となる．以上をまとめると，式(3)が得られる． (終わり)

成功確率の計算量：プラン $P=[O_1, \cdots, O_i]$ の成功確率は，$[O_1, \cdots, O_{i-1}]$ の成功確率の計算式を現在の時間で再計算した値(この計算は定数コスト)と，$Pr(+\langle s_i, t_i\rangle|\boldsymbol{e}_{i-1})$ の積であるから，成功確率はプランの長さとともに，インクリメンタルに計算できる．

次に，$Pr(+\langle s_i, t_i\rangle|\boldsymbol{e}_{i-1})$ の計算量を考える．式(3)の場合分けは，兄弟ノードをチェックするだけなので定数コストである．よって，式(3)全体の計算量は，定数オーダであり，さらに式(2)において，一つのオペレータの条件ノードは上限があるので，式(2)の計算も定数オーダである．その結果，成功確率は 1 ステップごとに定数オーダで計算可能である． (終わり)

著 者 紹 介

山田 誠二(正会員)

1984 年大阪大学基礎工学部卒業．1989 年同大学院博士課程修了．同年，大阪大学基礎工学部制御工学科助手．1991 年同大学産業科学研究所講師．1996 年 4 月より，東京工業大学大学院総合理工学研究科助教授，現在に至る．工学博士．人工知能，特に，動的環境への適応，ロボット学習，マルチエージェントの学習に興味を持つ．情報処理学会，日本認知科学会，AAAI，IEEE 各会員．