

インターリーブによるリアクティブ・プランニング

山田 誠二

大阪大学産業科学研究所
yamada@ai.sanken.osaka-u.ac.jp

プランニングと実行/観測を交互に行なうインターリーブ方式によって動的世界に対応可能なリアクティブ・プランニングの枠組みを提案する。外界の変化に依存したプランの成功確率を用いて、プランニングと実行/観測の切り換えタイミングの決定に理論的基盤を与える。成功確率は、オペレータの効果がどれだけ成功するかという確率とオペレーションに関係しない状態記述がどれだけ変化しないかという確率から、オペレーション間の因果関係に基づき計算される。さらに、成功確率をプランの探索のための評価関数として用いた探索アルゴリズムを示し、簡単な具体例により全体の動作を説明する。

Reactive Planning with Interleave

Seiji Yamada

ISIR, Osaka University
Mihogaoka 8-1, Ibaraki, 560 Osaka, Japan

For making a planning system adaptive to the dynamic world, various methods on reactive planning have been proposed. An interleave approach is a general framework in which planning and an execution/observation are mutually switched. However, it has a significant problem: when planning should be switched to an execution/observation, and few solutions has been proposed. In this paper, we propose a theoretical framework for an interleave approach in which the switching timing is determined with a success probability of a (partial) plan. The success probability is a possibility that a plan will be achieved successfully. In our method, planning is switched to an execution/observation only when a success probability of the plan decreases less than a threshold for an execution/observation. A success probability is also an evaluation function for searching a plan. Therefore, we present a new interleave planning algorithm with a success probability.

1. はじめに

元来プランニングは、人工知能において外界とのインタラクションを最も重要視する研究分野の一つである。しかし、従来のプランニング研究[1][2]は、外界への働きかけを保留し、外界の内部表現である信念(belief)の操作における探索問題の解決にそのほとんどの労力を費やしてきたと言えるだろう。このような方法論の根底には、完全なプランをいかに作るかが最重要であり、外界との関係において生じる様々な問題と記号操作のプランニングとは独立に考えてよいという前提がある。この前提は、記号操作のプランニング中に外界は変化しない、プランの実行は失敗しない等の非現実的な仮定のもとでは説得力のあるものとなるが、現実世界においてはむしろ先の仮定が満たされないことが常である。そのような現実世界において記号操作としてのプランニングは不可避免的に外界と干渉し、それ自体を独立に取り扱うことはもはや困難である。一般に、先の仮定を満たす外界を静的世界(static world)、満たさない外界を動的世界(dynamic world)と呼ぶ。

そこで、外界とのインタラクションを包括し、動的世界にも対応可能なプランニングであるリアクティブ・プランニング(reactive planning)の研究が近年活発になってきた[3]-[7]。本論文では、プランニングと実行を交互に行なうインターリーブ・プランニング[8]に対しプランの成功確率に基づく理論的枠組みを与え、実際のプランニング・アルゴリズムを提案し、最後に具体例を示す[10]。

2. リアクティブ・プランニング実現の方法論

リアクティブ・プランニングを実現するための方法として、これまでもいくつかの方法が提案されているが、主なものを以下に示す[9]。以降では、特に言明しないかぎり記号操作のプランニングを単に「プランニング」と呼ぶことにする。

- プランの選択：事前にプランを用意しておき、プランの探索をプランの選択に帰着し、即応化[7]。
- リアクティブ・エージェント：単純な刺激-反応エージェントの集合を用いる[3][5]。
- インターリーブ：プランニングと実行(観測)を交互に行なう[8][4]。

上記の方法はそれぞれ一長一短があるが、本研究ではメカニズムの一般性を考えてインターリーブを採用する。またさらに、実行/観測とプランニングというフェーズの設定下でもインターリーブを行わず、実行/観測とプランニングを並列に行なう方法が考えられる。しかし、そのような方法の問題点として、プランニングにより合目的性の高いプランを生成してから実行を開始することができないことがある。このことは、実行のコストあるいは実行の失敗修復のコストが非常に大きい場合などには致命的なものとなる。また、並列に行なう枠組みには外界の状況に応じて実行を停止しておく機能が欠けているのではないかということ、そしてそのような機能を付けていくと結局はインターリーブに近づいていくのではないかと思われる。

インターリーブ・プランニングとは、外界の変化に合わせてプランニングと実行/観測を交互に行うリアクティブ・プランニングの一方式である。直観的にも明らかなように、この方式ではプランニングと実行/観測の切り換えのタイミングをどのように決定するかが問題になるが、従来の研究では適切な解決がほとんどない。唯一とも言える解決案は、D. McDermot[8]によるもので、ワンステップ毎にプランニングと実行を繰り返すものである。しかし、本来は外界の変化が速ければ実行も速い時点で行ない、そうでなければゆっくりプランニングをした後で実行に移るとするのが望ましい。このような外界の変化の度合いに適応したプランニングと実行の切り換えタイミング決定の理論を含んだインターリーブ・プランニングの枠組みを本論文で提案する。また、実行とともに、ロボットにおいて非常にコストのかかる処理である観測のタイミングについても積極的に考慮する[11]。全体に理論的基盤がまだ脆弱なプランニング研究[12]において、インターリーブの理論的基盤を与えることを目指すのが本研究の1つの目的である。

3. プランの成功確率によるタイミング決定

3. 1 動的世界の定義づけ

まず、動的世界を定義づける。Fig.1に外界と信念の関係図を示す。Rは外界の状態を、S,S'はある行為主体AGがもつ信念を表し、OPは信念に対する記号操作(以降、オペレーション)、ACTは外界での物理的行為(以降、アクション)を表す。また、RからSへの矢印は、外界から信念への写像である。外界 R_1 を観測して得られた信念 S_1 に、オペレーション OP_1, \dots, OP_n を適用して得られた信念を S_i 、 R_0 に OP_1, \dots, OP_n に対応するアクション ACT_1, \dots, ACT_n を実行して得られた外界の状態を R_i 、そして R_i を観測して得られた信念を S'_i とする。以降の節でもこの表記を用いる場合がある。

また、すべての実線の矢印と波線の矢印は、それぞれ確定的写像と不確定的写像を表わす。さらに、目標達成とは、外界ではなく、信念において目標状態が満たされていることとする。この定義は、目標達成とは行為主体の主観的判断であり、行為主体にとって目標達成されていると判断されても、実際の外界においては目標状態が成立しておらず、そのため他者が観測すると目標達成されていない状況を許すことを意味する。

Fig.1でアクションが不確定的で、オペレーションが確定的とするのは妥当であろうが、ここで重要なことは、観測が行為主体にとって確定的であるとしている点である。一般に、あらゆる観測は誤差(不確定的要素)を含むと言われるが、本研究における観測の誤差とは、行為主体が得た観測結果の信念と真の信念(あるいは、より精度の高い信念)との差異を意味するとしてよい。しかし、その真の信念はどこから得られるのかというと、結局は行為主体以外の他者(典型的には、観測精度の高い人間の観測者あるいは神さま/仏さま)を想定して、その他者が外界を観測することにより得られた信念が行為主体に与えられると考えるしかない。しかし、真の信念が行為主体に与えられるのであれば、行為主体は最初からその真の信念を使えばいいのであり、その場合にはもはや観測の誤差はない。また、もし真の信念が行為主体に与えられないのなら、真の信念と行為主体の信念を比較可能である他者にとっては行為主体の観測の誤差が存在するかもしれないが、行為主体自身は自分の観測が誤差を含んでいるのか否かを判断する方法を持たないだろう。つまり、いずれの場合にせよ、本研究の対象とする状況においては、行為主体自身は観測の誤差を同定することも計測することも不可能であるというのが結論である。我々はこのような考察から、行為主体自身にとって観測は誤差を含まず確定的に正しいものとして扱うことにする。

まとめると、動的世界とは、観測された信念にオペレーションを行なうことによりえられる信念とそのオペレーションに対応するアクションを外界において行なった直後外界を観測して得られた信念との間の差異が生じるような外界を意味する。よって、Fig.1の形式においてAGに対する動的世界とは、 $S_i \neq S'_i$ となるような R_i を含む外界と定義する。また逆に静的世界とは、恒に $S_i = S'_i$ となる外界である。この定義による動的世界は、以下のような特徴を持つ。

- ・動的世界は、行為主体に依存して現出する。
- ・動的世界は、行為主体の信念の存在により始めて生ずる¹⁾。
- ・動的世界は、 S_{i+1} と S'_{i+1} との差異の原因に依存しない²⁾。

3. 2 切り換えタイミング決定の方針

インターリーブでは、プランニングと実行/観測の切り換えタイミングをどのような情報をもとに決定するかが問題となる。その問題に対する我々の直観的な回答は、次のようなものである。

「プランニングから実行への切り換えは、外界の変化が激しくて現在立てているプランが遂行不可能になる危険性が強まったときに起こる。」

この考え方に基づいて切り換えタイミングの決定法、さらにはプランニング全体を構成していくが、まずプランの遂行可能性をどのように表現するかである。

あるプランを動的世界で実行して目標となる状態が達成できるか否かは、必然的に不確定な要素を含んで

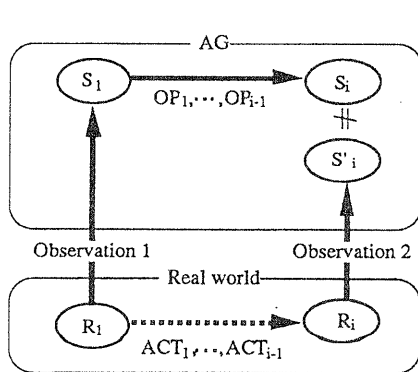


Fig.1 An dynamic world

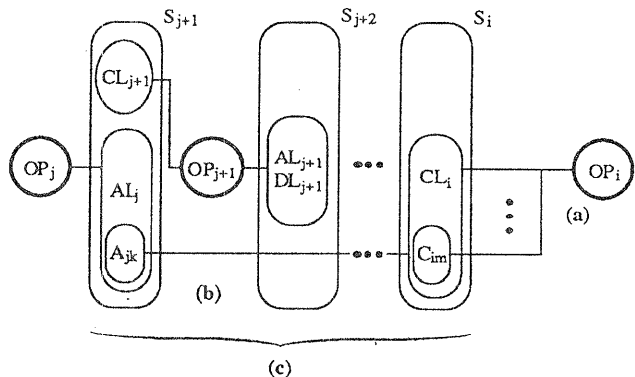


Fig.2 Causality of operators in a plan

¹⁾つまり、信念またはオペレーションを持たない、あるいは信念を予測しない行為主体にとって動的世界は存在しない。

²⁾原因が実行の失敗であっても、他のエージェントによるものであっても、さらには観測による信念と真の信念との差異であっても関係ない。

いる。そこでこのプランの不確定な遂行可能性をプランの成功確率という確率で表現する。ただし、この成功確率は、以下の条件を満たすものでなければならない。

(P1)外界の変化の程度を反映すること：外界の変化が激しい場合、当然プランの成功確率も低下する必要がある。

(P2)外界と独立なアクション自体の成功確率を含むこと：アクチュエータの故障等による成功確率の低下。

(P3)オペレーションの因果関係に基づく確率であること：一つのオペレーションが複数の結果をもたらす場合、後のオペレーションの適用条件であるものだけが、プランの成功確率に影響する。

以上の条件を満たすように、成功確率を設定していく。なお本研究では、プランナーに対し以下の条件を想定している。

- ・オペレータ：STRIPS-like[13]とする。条件リスト $CL_i=[C_{i1}, \dots, C_{ip}]$ ，削除リスト $DL_i=[D_{i1}, \dots, D_{iq}]$ ，追加リスト $AL_i=[A_{i1}, \dots, A_{ip}]$ で構成され，各リストの要素は正のリテラルのみである。また，条件リスト中のリテラルはすべて削除リストに含まれるとする。
- ・目標状態オペレータGOP：目標状態を条件リストとするオペレータで，このオペレータが適用されると目標が達成されたとする。削除，追加リストは空リストである。
- ・観測オペレータOPP：観測のオペレータをOPPとする。条件，削除リストは空で，追加リストは観測されたリテラルとする。
- ・信念：状態を正のリテラルで表現。
- ・プラン：信念とオペレーションの系列で， $[OPP, S_1, OP_1, S_2, \dots, S_n, OP_n, S_{n+1}]$ と表現される。あるいは，オペレーション系列だけの場合もある。

3. 3 適用確率によるプランの成功確率の定義

まず，プランの成功確率の定義づけを行なう。プランの成功確率とはプランの実行の成功確率を意味するが，プランの実行の成功とは何を意味するのであろうか。ここで，完全プラン(a complete plan)と部分プラン(a partial plan)という概念を考える。完全プランとは，目標状態オペレータをその最後にもつもので，目標状態を達成できるプランを意味する。また，部分プランとは目標状態オペレータを含まないプランのことで，それが実行されたとしても目標状態が達成されるわけではない。

プラン実行の最終的な目的は目標状態を達成することなので，完全プランにおいては目標状態が成り立つ確率，つまり目標状態オペレータが適用できる確率をプランの成功確率とみなすことができる。しかし，部分プランの場合，目標状態が部分的にしか，あるいは全く成り立っておらず，プランの実行が成功した状態の定義が難しい。もちろん，部分プランにおいてもサブゴールが明確に存在する場合はよいが，部分プランがサブゴールを含むタイミングで実行が始まる保証は全くない。よって，以下のように，我々は(サブ)ゴールに依存しないようにプラン実行の成功を定義づける。

「プランの実行が成功するとは，プラン中のすべてのオペレータが適用可能であることを意味する。」

この定義は，完全プランに対しては，目標状態オペレータが含まれているから，目標の達成に等しい。しかし，部分プランに関しては，とにかくプラン中のすべてのオペレータが適用されることが重要であるというヒューリスティックに基づいている。

このような定義のもとでプランの成功確率を計算するが，まずオペレータの適用確率を定義する。オペレータの適用確率は，プラン中のすべての $OP_i (1 \leq i \leq n)$ について割り当てられる。このプランのオペレーションに対応するアクション ACT_1, \dots, ACT_n を外界で実行し，それぞれのアクションの実行直後に観測により得られた信念を S'_2, \dots, S'_{n+1} とする。ある $S'_i (1 \leq i \leq n)$ に対し OP_i を適用するという試行をおこなう。この試行の結果得られる事象は，適用可能か不可能かのいずれかである。そこで適用可能である事象を $Ap(OP_i)$ ， OP_i の適用確率を $P(Ap(OP_i))$ とする。適用不可能な確率は， $1 - P(Ap(OP_i)) (\geq 0)$ である。

プランPlan中のオペレーション系列を OPP, OP_1, \dots, OP_n とする。このとき $Ap(OPP)=1$ であり，プランPlanの成功確率 $SP(Plan)$ は以下のようになる。

$$\begin{aligned} SP(Plan) &= P(Ap(OP_1) \cap Ap(OP_2) \cap \dots \cap Ap(OP_n)) \\ &= P(Ap(OP_1)) \cdot P(Ap(OP_2) | Ap(OP_1)) \cdot \dots \cdot P(Ap(OP_n) | Ap(OP_1) \cap \dots \cap Ap(OP_{n-1})) \\ &= \prod_{i=1}^n P(Ap(OP_i) | \bigcap_{j=1}^{i-1} Ap(OP_j)) \quad \dots \dots (1) \end{aligned}$$

よって，以前のオペレータがすべて適用可能であるという条件の下における条件つき適用確率

$P(\text{Ap}(\text{OP}_1)), P(\text{Ap}(\text{OP}_1)|\text{Ap}(\text{OP}_2)), \dots, P(\text{Ap}(\text{OP}_n)|\text{Ap}(\text{OP}_1) \cap \dots \cap \text{Ap}(\text{OP}_{n-1}))$ を求めればよい。以降でその算出法を説明していく。

3.4 プランの成功確率の計算

プランの成功確率の計算に必要な各種確率を定義する。

(1) リテラルの存在確率 $P(E(L, S_i))$

前述のプランの成功確率を求めるには、オペレーションによって得られる信念(Fig.1の S_i)中のリテラルが、観測によって得られる信念(S'_i)中に存在するかどうかを考えなければならない。そこで、リテラルの存在確率を以下のように定義する。

プラン中の任意の信念 S_i において、 S_i 中のリテラル L が信念 S_i 中でも成り立つことを「 S_i のリテラル L が存在する」と表現し、事象 $E(L, S_i)$ で表す。このとき、 $P(E(L, S_i))$ を S_i のリテラル L の存在確率とする。直観的には、信念中のリテラルが外界においても存在する確率を意味する。

(2) 先祖オペレータ $CP(C)$

プラン中のある具体化されたオペレータ OP_i の条件リスト中のリテラル C を信念に追加したオペレータをリテラル C の先祖オペレータと呼ぶ。 $CP(C)$ は、リテラル C の先祖オペレータを返す関数である。

(3) リテラルの追加確率 $P(E(A, S_{i+1})|\text{Ap}(\text{OP}_i))$

プラン中のあるオペレーション OP_i が信念 S_i に適用された結果、 S_{i+1} に追加されたリテラル A が、 $\text{Ap}(\text{OP}_i)$ を条件として存在する条件つき存在確率 $P(E(A, S_{i+1})|\text{Ap}(\text{OP}_i))$ を、オペレーション OP_i のリテラル A の追加確率とする。ただし、3.1の議論より観測(OPP)によって得られるリテラルの存在確率は1なので、 $P(E(A, S_i)|\text{Ap}(\text{OPP}))=1$ (A は任意)とする。また、 OP_i の追加確率は OP_i 以外の適用確率と独立である。

この追加確率は、オペレータの削除リスト中のリテラルには割り当てられない。これは、オペレータが条件リスト中に負のリテラルを持たないため、プランの成功確率の計算に削除が成功するか否かが無関係なためである。

(4) 状態不変確率 $P(\text{NC}(\text{Pre}, T))$

ある時点で外界で存在していて且つオペレーションにより削除されないリテラル L (述語記号 Pre)が、 $T(\text{sec})$ 後も外界で存在している確率を状態不変確率 $P(\text{NC}(\text{Pre}, T))$ とする。この確率はオペレーションの適用確率とは独立とする。事象 NC のパラメータが述語記号と時間なのは、状態不変が対象から独立であり対象間の関係と経過時間にのみ依存するという仮定による。

以上の確率のうち、追加確率と状態不変確率がプランの成功確率を計算するために使われる。

(5) プランの成功確率

Fig.2は、プランのオペレーション間の因果関係を表すネットワークである。 OP_i の条件つき適用確率 $P(\text{Ap}(\text{OP}_i)|\text{Ap}(\text{OP}_1) \cap \dots \cap \text{Ap}(\text{OP}_{i-1}))$ の計算法をFig.2で説明する。この適用確率は、 $\text{Ap}(\text{OP}_1) \sim \text{Ap}(\text{OP}_{i-1})$ がすべて起こる条件下で OP_i の条件リスト中のリテラル $C_{ik}(1 \leq k \leq p)$ がすべて S_i で存在する確率(Fig.2の(a))であり、さらにそれぞれのリテラル C_{ik} が条件 $\text{Ap}(\text{OP}_1) \cap \dots \cap \text{Ap}(\text{OP}_{i-1})$ の下で存在する事象 $E(C_{ik}, S_i) \sim E(C_{ip}, S_i)$ はお互い確率的に独立とする。よって、下式が成り立つ。

$$\begin{aligned} & P(\text{Ap}(\text{OP}_i)|\text{Ap}(\text{OP}_1) \cap \dots \cap \text{Ap}(\text{OP}_{i-1})) \\ &= P(E(C_{i1}, S_i) \cap \dots \cap E(C_{im}, S_i)|\text{Ap}(\text{OP}_1) \cap \dots \cap \text{Ap}(\text{OP}_{i-1})) \\ &= \prod_{m=1}^p P(E(C_{im}, S_i)|\text{Ap}(\text{OP}_1) \cap \dots \cap \text{Ap}(\text{OP}_{i-1})) \quad \dots\dots(2) \end{aligned}$$

次に、 $P(E(C_{im}, S_i)|\text{Ap}(\text{OP}_1) \cap \dots \cap \text{Ap}(\text{OP}_{i-1}))$ を求める。この確率は、オペレータ $OP_1 \sim OP_{i-1}$ がすべて適用可能である条件下で、信念 S_i においてリテラル C_{im} が存在する条件つき確率である。リテラル C_{im} は祖先オペレータ $CP(C_{im})=OP_j$ により信念 S_{j+1} に初めて追加されたものであり、 C_{im} は $OP_{j+1} \sim OP_i$ が適用される間(Fig.2の(c))はオペレーションに関与しないので状態不変確率(例えば、Fig.2の(b))に従う。また、祖先オペレータが観測オペレータの場合、観測開始からこの成功確率計算の開始までの時間と $OP_1 \sim OP_{i-1}$ の実行時間の合計との和を時間のパラメータとした状態不変確率に従う。よって、 S_i における C_{im} の存在確率は、下式により得られる。

$$\begin{aligned} & P(E(C_{im}, S_i)|\text{Ap}(\text{OP}_1) \cap \dots \cap \text{Ap}(\text{OP}_{i-1})) \\ &= \begin{cases} P(\text{NC}(\text{Pre}(C_{im}), T + \sum_{k=1}^{i-1} T(OP_k))) & (OP_j = \text{OPP}) \quad \dots\dots(3) \\ P(E(C_{im}, S_{j+1})|\text{Ap}(\text{OP}_j) \cdot P(\text{NC}(\text{Pre}(C_{im}), \sum_{k=j+1}^{i-1} T(OP_k)))) & (OP_j \neq \text{OPP}) \end{cases} \quad (5) \end{aligned}$$

ただし上式で、 T は観測開始を0(sec)とした実時間軸上での成功確率計算の開始時刻(sec)である。また、 $T(OP_i)$ は OP_i の実行に要する実時間(sec)である。最終的に、(1), (2), (3)式よりプランの成功確率が得られる。SP(Plan)は、以下の入力から計算可能である。なお、具体例は§5で示される。

・追加確率 ・状態不変確率 ・アクションと観測に要する時間 ・プラン

このプランの成功確率が、3.2の成功確率が満たすべき条件(P1)~(P3)を満たすことを確かめておこう。まず、状態不変確率はいうに及ばず追加確率も、外界の変化速度を反映して決定された値が入力されるので、(P1)が満たされる。また、追加確率は、例えばマニピュレータの故障等に起因する外界の変化と独立にアクション自体が失敗する確率を含むので、(P2)が満たされる。また、リテラル単位の因果関係に基づき計算されるので、(P3)も満たされる。

3. 5 実行/観測のタイミング決定

前節で得られたプランの成功確率により、実行/観測のタイミング決定を行う。基本的には、プランニングを進めていき、そこで得られる最適なプランの成功確率がしきい値より下回ると実行/観測を行う。ここでしきい値は、実行のしきい値と観測のしきい値の2つが独立に与えるものとする。

プランの成功確率は基本的に確率の積なので、観測が行われたとき以外はプランが大きくなるにつれて成功確率は単調減少する。観測直後には、成功確率は増加する。実行と観測のしきい値の設定方法、2つのしきい値間の定量/定性的関係については、現在のところ検討中であるが以下のような傾向が考えられる。

- ・実行/観測のしきい値は、実行/観測のコストに依存する。コストが大きいとしきい値は低く設定され、コストが小さいとしきい値は高くなる。人間は観測のコストが非常に少ないため、観測のしきい値は低く設定され、その結果観測を常に行なうことができるのに対し、ロボット等ではその逆である。
- ・再プランニングのコストが大きいと、実行のしきい値は低い。

4. 成功確率によるプランニング・アルゴリズム

本節では、プランの成功確率を用いたインターリーブ・プランニングの全体的なアルゴリズムを説明する。インターリーブ・プランニングの場合、全体のアルゴリズムは必然的に、記号操作のプランニングと実行/観測というアクションの双方を組み込んだものとなる。前節である一つのプランが与えられれば、その成功確率を計算し、プランニングと実行/観測のタイミングを決定する方法は得られたが、全体のプランニングではその成功確率を評価関数として用いてプランを探索する手続きが必要となる。

本研究における探索手続きは、

「ヒューリスティックによる枝がりをを用いた横型前向き探索を行ない、同じステップ数のすべてのプラン候補の成功確率がしきい値よりも下がったとき、それらの中で評価関数の最高値をもつ最適プランを実行し観測を行なう」

というものである。詳細なアルゴリズムの記述は、Fig.3に示される。このような手続きを設定した理由を説明する。

(1) 前向き探索

インターリーブ・プランニングでは、動的世界において部分プランの実行を行なわなければならないため、生成されるあらゆるプランは初期信念(あるいは外界)に適用できるものでなければならない。よって、前向き探索を採用した。

(2) 横型探索

プランの成功確率は単調減少するので縦型探索あるいは最良優先探索を行なうと、最初にしきい値を下回ったプランが実行され、最適プランが得られず好ましくない。よって、横型探索を行ない、あるレベル(プランのステップ数)におけるすべてのプランの成功確率がしきい値を下回ったとき、評価関数の最も高いプランを実行する。なお、分岐限定法等でも同様の探索が可能と考えられるが、アルゴリズムの単純さを考えて横型探索を採用した。また、明らかに無駄なプランの探索を行なわないように、ヒューリスティックによる枝刈りを行っている。

(3) 評価関数

実行開始が決定したとき、複数の候補から評価関数により最適プランを選択するが、その評価関数は f (Plan) = U (Plan) · SP(plan) とする。この f は、意志決定理論(decision theory)でも用いられるもので、 U はPlanが実行されたことにより得られる効用(utility)の関数である。典型的には、目的-手段解析等における目標との差異などが考えられ、経験的にわかっていない場合は定数にしておく。また、プランの実行/観測タイミング決定において成功確率のかわりに f を使うことも考えられるが、タイミング決定が有限時間で可能な

ことが関数が単調減少するという性質に依拠しているので、単調減少が保証されていない関数 f を用いることはできない。

以上の手続きにより実行すべきプランが見つかり、そして実行が行なわれる。このとき、観測のしきい値を下回ったプラン全体を実行した直後に観測が行なわれる。次節で具体例を示す。

5. 具体例

タイミング決定を伴ったインターリーブ・プランニング全体の動作例を示す。いま、Table1に示すようなオペレータとそれぞれの入力確率を考える。オペレータは、[14]から引用した。Timeはアクションを外界において行なった場合に要する時間(sec)を表し、ここでは状態不変確率を次式を用いてTimeから計算している。これは時間に比例して外界が変化することを意味する。また簡単のため、状態不変確率は述語記号に依存しないとした。

$$P(\text{NC}(\text{Pre}(L),T)) = \begin{cases} 1 - \frac{T}{200} & (T \leq 200) \\ 0 & (T > 200) \end{cases}$$

ここでは、putdownの追加確率を0.5とし、さらに外界においては定期的に外乱が発生することによって動的世界を実現している。他のパラメータは、以下のように設定した。

```

GOAL: Goal state, Ob: an observation,  $f(\text{Plan}) = U(\text{Plan}) \cdot \text{SP}(\text{Plan})$ 
Exe(NODE) : an execution of the plan in NODE
N_LIST : a list of nodes in a current level
SP_max_node(N_LIST) : the node with a maximum value of SP(Plan) in N_LIST
 $f_{\text{max\_node}}(N\_LIST)$  : the node with a maximum value of  $f(\text{Plan})$  in N_LIST
SP(N) : a success probability of a plan in node N
Plan(N) : a plan in node N
Comp_plan(N_LIST) : a list of all complete plans in N_LIST
Planning(N_LIST) : a function returning next level N_LIST after pruning with heuristics
NODEi=[So, Si, [OP1,..., OPi], P(S(PLAN))]

procedure Interleave-planning :
begin
  S0 ← Ob;
  N_LIST ← [[S0,S0,[ ],P(OB)]];
  while S0 ⊇ GOAL do
    begin
      N_LIST ← planning(N_LIST);
      if Comp_plan(N_LIST) ≠ [ ] then
        begin
          Exe(SP_max_node(Comp_plan(N_LIST)));
          S0 ← Ob;
          N_LIST ← [[S0,S0,[ ],P(OB)]]
        end
      else
        begin
          MAX_SP ← Sp(SP_max_node(N_LIST));
          if MAX_SP ≤ an execution threshold then
            begin
              Exe( $f_{\text{max\_node}}$ (N_LIST));
              N_LIST ← [ $f_{\text{max\_node}}$ (N_LIST)]
            end
          if MAX_SP ≤ an observation threshold then
            begin
              S0 ← Ob;
              N_LIST ← [[S0,S0,[ ],P(OB)]]
            end
          end
        end
      end
    end
  end
end

```

Fig.3 An algorithm of interleave planning

- 観測に要する時間：5 (sec)
- プランニングに要する時間：プランのステップ数 (sec). よって, ステップ数nのプランの成功確率計算開始時間 $T' = n+5$ となる.
- 実行/観測のしきい値：ともに0.6
- 外乱の発生周期：45 (sec)
- 枝刈り用ヒューリスティック：ループの禁止, 目標順序(goal ordering)の指定
- 効用関数：定数とする. つまり, 成功確率だけで評価される.

Fig.4のような積み木の世界の問題を考える. このとき, 枝刈りによりFig.5に示されるP1, P2の2つのプランだけが残り, 探索空間を形成する. これらのプランは, unstack(a,b)により持ち上がられた積み木aを机の上に置くか(putdown(a)), 別の積み木の上に積むか(stack(a,d))の違いがある. 成功確率という概念を用いない従来のプランニングでは, この2つのプランに優劣をつけることは困難であり, 実際にはputdown(a)の実行に失敗する危険性の大きいP2が選択されてしまう可能性があるが, 成功確率により安全なP1が選択される.

我々の方法では, 各レベルにおいてプランの成功確率が計算されるが, 例としてFig.5の最適プラン上の部分プランである[unstack(a,b),stack(a,d)]をFig.6の因果ネットワークで成功確率の計算を説明する. まず, $P(E(C_{im}, S_i) | Ap(OP_1) \cap \dots \cap Ap(OP_{i-1}))$ が式(3)より下のように得られる. ただし, 観測に5(sec)かかり, さらに2ステップのプラン生成後なので $T' = 5+2=7(sec)$, またTable1より $T(unstack(a,b))=10(sec)$ となる.

$$P(E(clear(a)S_1)) = P(NC(clear,7)) = 1 - \frac{7}{200} = 0.965$$

ここでは状態不変確率は述語記号によらず一定値なので式(2)より,

$$P(Ap(unstack(a,b))) = P(NC(clear,7))^3 = 0.965^3 = 0.90$$

が得られる. 同様に式(3), (2)より, 以下の値が得られる.

$$P(E(holding(a)S_2) | Ap(unstack(a,b))) = P(E(clear(d)S_2) | Ap(unstack(a,b))) = 1.0$$

pickup(X) :	
cond&delete-list	[ontable(X), clear(X), handempty]
add-list	[holding(X)]
putdown(X) :	
cond&delete-list	[holding(X)]
add-list	[ontable(X), clear(X), handempty]
stack(X, Y) :	
cond&delete-list	[holding(X), clear(X)]
add-list	[on(X,Y), clear(X), handempty]
unstack(X, Y) :	
cond&delete-list	[on(X,Y), clear(X), handempty]
add-list	[holding(X), clear(X)]

Operators	Time	Ai	Addition Pr.	NCP
pickup(X)	7.00	holding	1.00	0.96
putdown(X)	5.00	ontable(X)	0.50	0.97
		clear(X)	0.50	
		handempty	0.50	
stack(X,Y)	10.00	handempty	1.00	0.95
		on(X,Y)	1.00	
		clear(X)	1.00	
unstack(X,Y)	10.00	holding(X)	1.00	0.95
		clear(Y)	1.00	

Table1 Input operators and probabilities

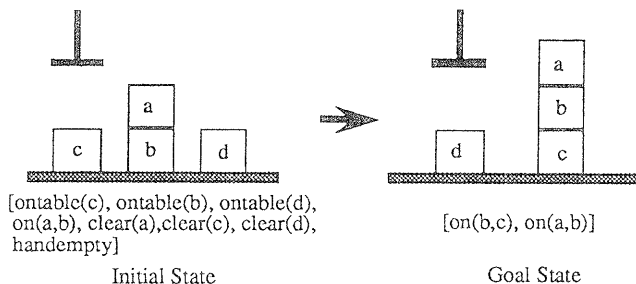


Fig.4 A problem

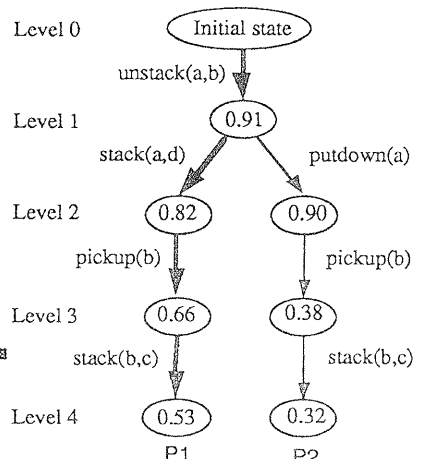


Fig.5 A part of searching

$$P(E(\text{clear}(d)S_2) | \text{Ap}(\text{unstack}(a,b))) = P(\text{NC}(\text{clear}, T^+ + T(\text{unstack}(a,b)))) = 0.915$$

$$P(\text{Ap}(\text{stack}(a,d)) | \text{Ap}(\text{unstack}(a,b))) = 1.0 \times 0.915 = 0.915$$

よって、下の成功確率が得られる。

$$\begin{aligned} SP(\text{unstack}(a,b), \text{stack}(a,d)) &= P(\text{Ap}(\text{unstack}(a,b)) | \text{Ap}(\text{unstack}(a,b))) P(\text{Ap}(\text{stack}(a,d)) | \text{Ap}(\text{unstack}(a,b))) \\ &= 0.915 \times 0.90 = 0.82 \end{aligned}$$

Table2とFig.7に最適プランの成功確率を示す。ところで、Fig.5の探索において成功確率を評価関数とした縦型で探索すると、レベル2でP2が選ばれてしまう。このような誤った選択を避けるためにも横型探索が有効である。最終的に得られるインターリーブの動作結果がFig.8である。まず、4ステップのプランを立てたときに成功確率がしきい値を下回り、実行が行なわれ、その直後に観測がなされる。ところが、45(sec)において外乱1が発生し積み木bが落ちてしまう。すると再び観測された信念を初期状態として再プランニング(planning2)が行なわれる。そして、またステップ4で実行が開始されるが、ここでも外乱2が発生している。よって、再プランニング(planning3)が行なわれ、完全プラン[pickup(a),(stack(a,b),GOP)]が見つかり、即座に実行される。この実行は次の外乱発生の前に終了するので、めでたく目標達成が実現する。

もし、この問題をインターリーブを行なわない単純な前向き横型/縦型探索でプランニングしようとする、常に完全プランの生成そして実行を行なう分だけ、目標達成までの時間がかかると思われるが、組織的実験及び解析は今後の課題である。

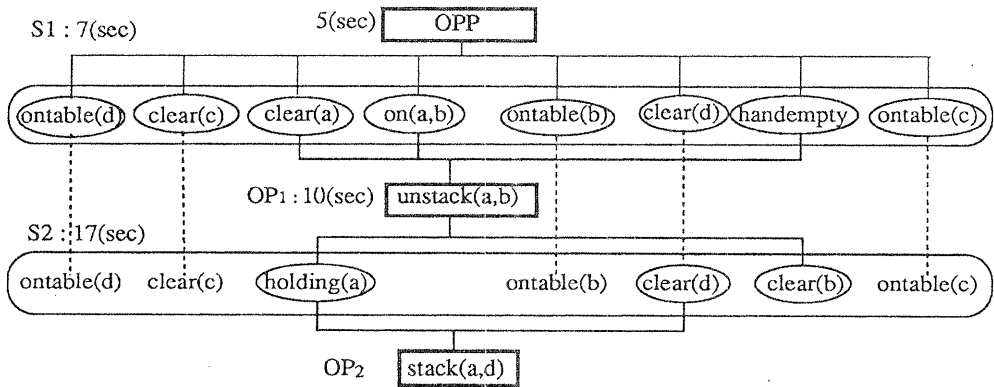


Fig.6 Causality of a subplan [unstack(a,b), stack(a,d)]

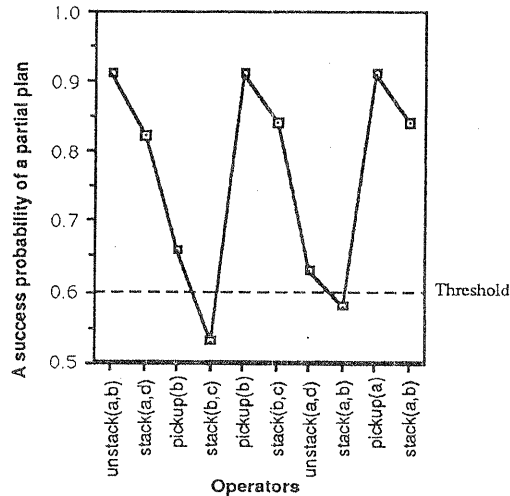


Fig.7 A behavior of a success probability

6. まとめと今後の課題

従来の研究では明確に決定されていなかったプランニングと実行/観測の切り換えタイミングを、プランの成功確率を基に決定する方法を提案した。追加確率、状態不変確率は、訓練期間を設けて、その間に実際に各オペレーションの行為を実行し、経験的に求めることを考えている。実行/観測のしきい値の決定方法及び関連研究との比較は、今後の課題である。

<謝辞>

日頃お世話になっている豊田順一教授に感謝いたします。また、本研究のインプリメント、§5の内容について大きく貢献してくれた磯田佳憲君と有益なご指摘を受けた淡誠一郎助手に感謝する次第です。

<参考文献>

- [1] 安部：プランニング，人工知能学会誌，5(6)，pp.737-747 (1990)
- [2] Hendler, J., Tate, A. and Drummond, M.: AI Planning Systems and Techniques, *AI Magazine*, vol.11, no.2, pp.61-77 (1990)
- [3] R. J. Firby, An investigation into reactive planning in complex domain, in: *Proceedings AAAI-87*, (1987) 202-206
- [4] Durfee, E.H. and Lesser, V.R.: Incremental planning to control a blackboard-based problem solver, *In Proceedings of the Fifth National Conference on Artificial Intelligence*, pp.58-64 (1986)
- [5] Georgeff, M.P. and Lansky, A.L.: Reactive reasoning and planning, *In Proceedings of the Sixth National Conference on Artificial Intelligence*, pp.677-682, (1987)
- [6] Vere, S.A.: Planning in Time: Windows and Durations for Activities and Goals, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5, no.3, pp.246-267 (1983)
- [7] Schoppers, M.J.: Universal Plans for Reactive Robots in Unpredictable Environments, *In Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pp.1039-1046 (1987)
- [8] Mc Dermott, D. : Planning and Action, *Cognitive Science* 2, pp.71-110 (1978)
- [9] M. Marks, K. J. Hammond and T. Converse, Planning in an open world: a pluralistic approach, in: *Proceedings Case-Based Reasoning Workshop, Florida* (1988) 271-285
- [10] 磯田，山田，豊田：インターリーブによるリアクティブ・プランニングの枠組み - プランの評価とプランニング -，第43回情報処理学会全国大会 (1991)
- [11] 三浦，白井：不確かさを考慮した視覚と行動のプランニング，第5回人工知能学会全国大会，pp.749-753 (1991)
- [12] McDermott, D.: Robot Planning, *In Proceedings of the Ninth National Conference on Artificial Intelligence*, pp.930-931 (1991)
- [13] Fikes, R.E. and Nilsson, N.J.: STRIPS: A New Approach to the Application of Theorem Proving, *Artificial Intelligence* 2, pp.189-208 (1971)
- [14] Nilsson, N. J. : 人工知能の原理，pp.248-291 (1980)

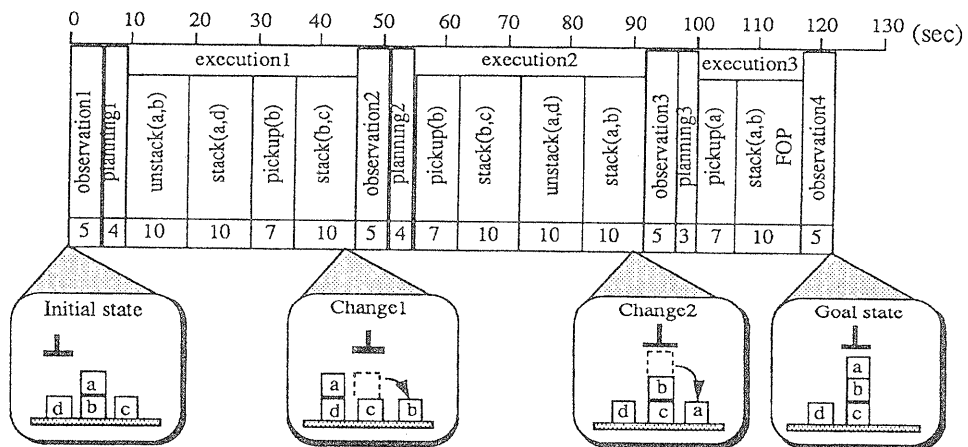


Fig.8 A obtained plan on the time axis