

# プランニングと行動の一貫性に基づく移動ロボットの プランニングと行動の並列スケジューリング

## Parallel Scheduling of Planning and Action of a Mobile Robot Based on Planning-Action Consistency

三浦 純  
Jun Miura

大阪大学大学院工学研究科電子制御機械工学専攻  
Dept. of Computer-Controlled Mechanical Systems, Osaka University  
jun@mech.eng.osaka-u.ac.jp, <http://www-cv.mech.eng.osaka-u.ac.jp/~jun/>

白井 良明  
Yoshiaki Shirai

大阪大学大学院工学研究科電子制御機械工学専攻  
Dept. of Computer-Controlled Mechanical Systems, Osaka University  
shirai@mech.eng.osaka-u.ac.jp, <http://www-cv.mech.eng.osaka-u.ac.jp/~shirai/>

**Keywords:** parallel scheduling of planning and action, planning-action consistency, planning under uncertainty, planning cost, mobile robot

### Summary

This paper proposes a novel method to schedule parallel execution of planning and action of a mobile robot. The method considers the following two types of parallelism. (1) *acting while planning*: if a partial planning result can be used to determine feasible actions, such actions can be executed while the planning process is still going. (2) *planning while acting*: if the result of the current action is predictable, planning for the next action can start in advance of the completion of the current action. The proposed method uses the notion of *planning-action consistency* to guide the scheduling. The method has been successfully applied to a mobile robot navigation problem under sensor uncertainty.

### 1. はじめに

実世界で動作するエージェントのプランニング(行動計画の自動生成)において, 資源の有限性および不確かさは大きな問題である. 不確かさの下でのプランニングでは, 起こり得る複数の可能性を考慮する必要があり一般にコストがかかるので, 資源の有限性がより厳しい拘束となる. プランニングのコストを陽に考慮することによりプランニングを制御することは, 全体の効率を上げるための1つの方法である [Russell 91, 三浦 98] が, 行動とプランニングを並列実行することによってさらに効率を向上させることができる. 本論文では, プランニングと行動の依存関係を考慮してそれらを適切に並列スケジューリングする手法を提案する. 例題として, プランニングと視覚認識および移動が並列実行できる移動ロボットのナビゲーション問題 [Miura 97] をとりあげる.

図1に対象とするプランニング問題の例を示す. 移動ロボットは環境中の物体の形状やおよその位置を記述した地図を与えられて, 障害物を避けながら指示された目的地 (*destination*) へ向かう. 狭い空間 (ゲートと呼ぶ) を通る経路があるが, 地図には正確な位置は記述されておらず, また視覚情報の不確かさのため, ゲートを通れる

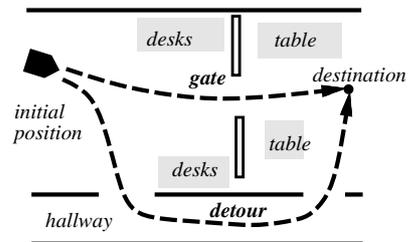


図1 プランニング問題の例

かどうかは現在の位置 (*initial position*) からの観測ではわからないとする. ロボットは必要であればステレオ視覚でゲートを観測し通過可能性を調べる. 廊下を通る迂回路は通れることはあらかじめわかっているが, 目的地までの距離は長い. ロボットは自身の移動量を知ることができ移動誤差はないものとする. ここでのプランニングの目的は, 視覚のコストと不確かさ, およびプランニングのコストを考慮して, ロボットを効率よく目的地へ誘導するための観測地点の系列を生成することである. プランニングの際に考慮すべき不確かさはゲート幅の不確かさだけである.

このプランニング問題は, 次の2つの副問題に階層的に分割できる:

- 上位のプランナが副目標 (観測地点) を生成する .
- 下位の動作コントローラがその副目標を達成するための行動 (移動と観測) を選択し, 実行する .

本論文ではこのような階層構造をもつプランニング問題を対象として, プランニングと行動の並列スケジューリング手法を提案する .

## 2. プランニングと行動の並列性の 2 つの観点

本章では, プランニングと行動の並列性を, 行動しながらのプランニング (planning while acting) とプランニングしながらの行動 (acting while planning) という 2 つの観点から分類する .

まず, プランニングと行動が逐次的に行われる場合を考える . このとき, プランニングの結果得られた行動を実行し, その行動の実行結果に基づいて次のプランニングが行われる, という手順が繰り返される . したがって, 並列化のためには互いの結果を予測し, その予測に基づいて早めに処理を開始することが必要である . どちらの結果を予測するかによって上述の 2 つの並列化が考えられる .

### 2.1 行動しながらのプランニング

現在実行中の行動の結果を予測し, その行動が終了する前に次の行動のプランニングを並列に実行することを, 行動しながらのプランニングと呼ぶ . 実時間探索 [Korf 90] やインターリーブ [Nourbakhsh 97] はこの並列性を実現するのに適している . すなわち, ある行動を実行中に, その行動の結果得られる状態に対する最適な行動をプランニングできる . これを図式的に表したのが図 2(a) である . 動作コントローラはプランナから指示された行動を実行し, それが完了すれば次の指示が来るまで待つ .

継続計算 (continual computing) [Horvitz 97] は, 時系列的に発生する問題群を予測し, 起こりそうな問題に

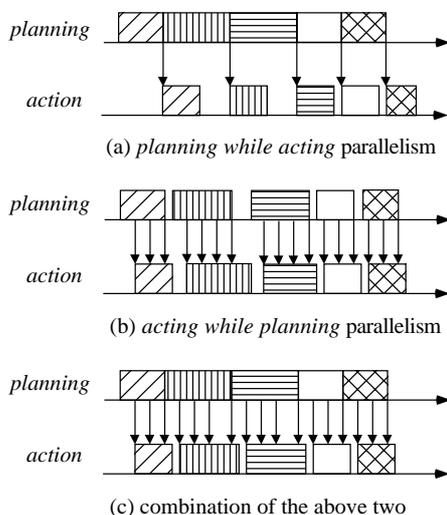


図 2 2 つの並列性とそれらの組み合わせ

対するプランニングをあらかじめ行っておくことにより, 問題発生待ち時間の効率的な利用を目指すものである . この考え方を行動しながらのプランニングの枠組で考えると, 現在実行中の行動の帰結が確率的にのみ予測できる場合に相当する .

### 2.2 プランニングしながらの行動

プランニングの途中結果から実行可能な行動を選択し, プランニングが完全に終了する前に行動を開始することをプランニングしながらの行動と呼ぶ (図 2(b) 参照) . この並列性はいままでほとんど研究されてこなかったが, 特に不確かさのためにプランニングコストが高い場合に, 全体の効率を向上させるために重要となる .

## 3. プランニングと行動の一貫性に基づくスケジューリング

本章は上述の 2 種の並列性を両方実現するための基本方針を述べる (図 2(c) 参照) . スケジューリングにあたっては, 並列実行することのできるプランニングと行動の組を決定する必要がある . そのために, プランニングと行動の一貫性 (planning-action consistency) という考え方を導入することを提案する .

### 3.1 プランニングしながらの行動

1. で述べた問題の階層的分解において, 上位レベルの副目標のプランニング過程は, 初期の副目標候補集合から徐々に候補を絞っていき, 最終的に 1 つの副目標を選択する過程と見ることができ [Kambhampati 95] . 途中の候補集合と一貫性のある行動があれば, それはプランニングしながら実行できる .

以上の考え方を図 3 を用いて説明する . まず初期の副目標候補集合 (feasible subgoal space (FSS) と呼ぶ) を生成する (図 3(a) 参照) . 次に FSS と一貫性のある行動の集合 (consistent action space (CAS) と呼ぶ) を生成し (例えば, 図 3(a) の網掛領域), CAS から一番よいものを選んで実行する (図 3(b) 参照) . プランニングが進むにつれ FSS は減少し, 同時に一貫性のある行動が繰り返し選択され実行される (図 3(c) 参照) . 最後に, 1 つの副目標が選択され, その少し後にロボットはそこへ到達し観測する (図 3(d) 参照) . もしプランニングと行動を逐次的に実行するならば, 副目標を決定したときロボットはまだ初期位置に留まっている . したがって, プランニング開始時から副目標を決めるまでにロボットが進んだ距離が, 並列実行により得られる利得である .

プランニングと行動は非同期並列に実行可能なので, 可能なスケジュールの空間は膨大になる . そこで, 行動を変えるタイミングを, プランニングの候補を絞る繰り返し処理の, 各回の繰り返し終了時に限定する .

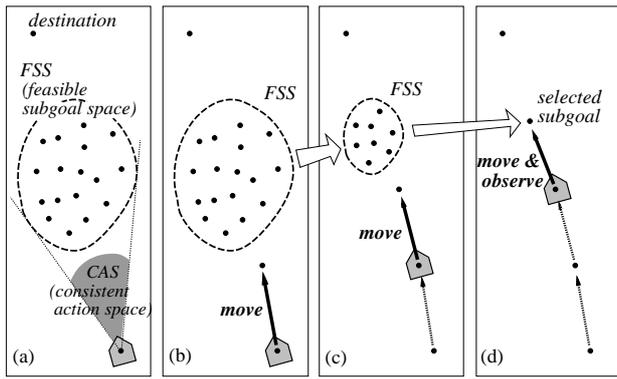


図3 プランニングしながらの行動の例

以上のプランニングしながらの行動のスケジューリングのためには、まず

- FSS と行動の一貫性を評価する基準 (一貫性基準と呼ぶ)

が必要である。さらに

- 副目標候補を繰り返し絞ってゆくプランナ、および
- いくつかの繰り返しを終了し候補を1つ選択するかを決める基準 (決定基準と呼ぶ)

が必要である。これらは一般に、問題ごとに設定する。

### 3.2 行動しながらのプランニング

ここでは、1つの副目標を決めてからロボットがそれを達成するまでの時間 (図3(d)参照) の有効利用を考える。現在の行動の起こり得る結果の集合が予測可能かつ列挙可能であるとする\*1。ある予測される状態に対してその次の行動をプランニングすること (プランニングオプションと呼ぶ) は、現在の行動に対し一貫性があると考えられることができる。そこで、現在実行中の行動が終了するまでの間に、適切なプランニングオプションの集合を選択し、実行する。

行動しながらのプランニングのための時間が予測できる場合には、最適なプランニングオプションの集合を求め実行する。予測できない場合には、もっとも有効だと思われるプランニングオプションから順番に実行してゆく。われわれのプランニング問題の場合は前者であるので、プランニングオプション選択に線形計画法などの最適化手法が適用できる。また Contract anytime algorithm[Zilberstein 93]に基づくプランニング手法も適用可能である。

## 4. プランニングアルゴリズム

### 4.1 プランの表現

行動と状態を次のように定義する。行動とは、現在の位置から次の観測位置までの移動とそこでの観測との

\*1 例えば、行動の確率モデルが与えられれば、行動終了後の状態が確率的に予測できる。

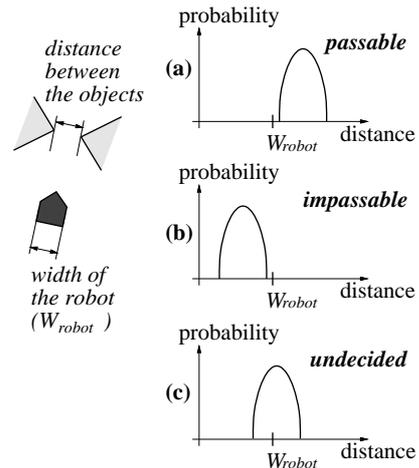


図4 ゲートの3つの状態

組である。状態とは、現在のロボットの位置とゲートの幅の見積り値である。観測には不確かさがあるので、ロボットはゲート幅の確率分布を得る [Miura 97]。

ある観測の後で、ゲートの状態を次の3つのどれかに分類する (図4参照)。ロボットの幅がゲート幅の確率分布の最小値より小さい時には、ゲートは通過不可能、ロボット幅が確率分布の最大値より大きい時には通過可能、その他の場合には、通過可能性は未定である。

ある行動後の状態は観測結果に依存し、あらかじめ決定的に予測することはできないが、プランニング時には、起こり得るすべての状態について副プランを生成し、それをもとにプランを評価する必要がある。したがって、プランの探索木は AND-OR 木となる。OR ノードは一つの選ばれた行動を表し、AND ノードは行動の結果得られる可能な状態の一つを表す。また、得られたプランは1つの AND ノードに対する OR ノードが1つだけの、特殊な AND-OR 木となる (図5参照)。プランの質はその実行コストで評価する。ここでは、実行コストとして観測と移動にかかる時間の期待値を用いる。

AND-OR 木の葉は、終端ノードが開ノードである。終端ノードでは、ゲートの通過可能性は定まっている。すなわちゲートを通るか迂回路をとるかのどちらかに定まっている。開ノードでは通過可能性未定であり、そこから次にとる行動は定まっていない。プラン候補はその開ノードのうちの一つを選んで展開する (そのノードに対する副プランを生成する) ことにより、精密化される。開ノードを展開する際には、ゲート幅の予測される不確かさの範囲にある固定した粒度で均等に離散化し、離散化された各状態に対して副プランを生成する。

### 4.2 繰り返し改善としての定式化

本論文では、プランニング過程を任意時間繰り返し改善 (anytime iterative refinement) [Boddy 89] の形に定式化する。これにより、徐々に候補集合が絞られていく過

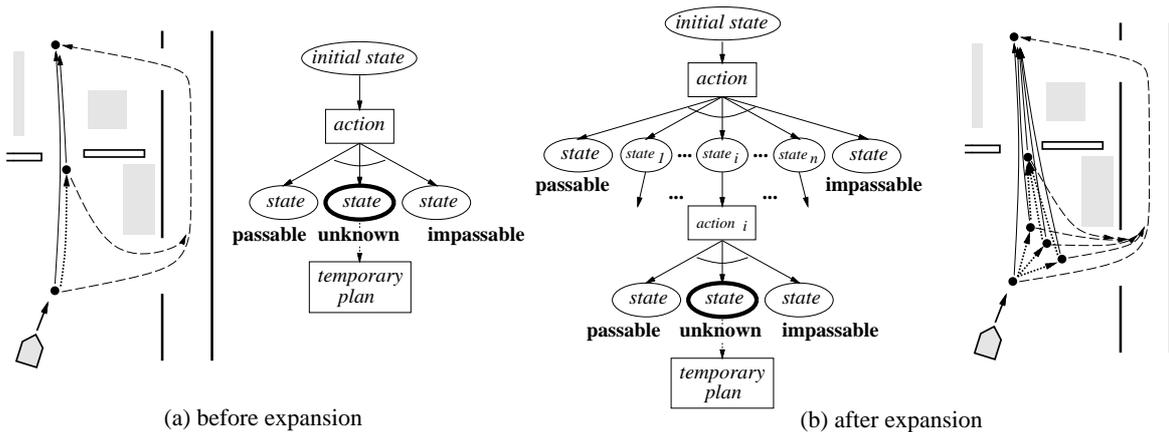


図5 プラン候補の開ノードの展開前後の探索木と対応する状況

程を実現できる。

繰り返し改善では、実行可能プラン (feasible plan) の空間でプランを探索する。ここで、実行可能プランとは、プラン実行中に生じ得るすべての状態に対する行動が定められているものをいう。そこで、すべての開ノードに一時的に実行可能プラン (暫定プランと呼ぶ) を割り当てておき、必要に応じてこの一時的な実行可能プランをよりよい実行可能プランに修正することにより、プラン候補を精密化する。

暫定プランは容易に計算できる必要があるので、次の実行可能プランを暫定プランとして採用する。

現在位置からゲート直前の位置\*<sup>2</sup>へゆき、観測する。ゲートが通過可能ならゲートを通って目的地へ向かう。通過不可能なら迂回する。

プラン候補のすべての開ノードに、以上の暫定プランを与えることにより、プラン候補のコスト (暫定コストと呼ぶ) が計算できる。

プラン候補の精密化 (展開) は次のように行う。展開前には、ある未定ノードは1つの開ノードとして扱われ、1つの暫定プランが付加されている (図5(a) 参照)。これを展開するには、まず適当な粒度で不確かな状態を離散化し、各離散化された状態に対して1段の探索を行って次の最適な観測位置を求め、それでも残った未定状態 (開ノード) に再び暫定プランを割り当てる (図5(b) 参照)。なお、図5(a) は初期暫定解の例を示していることに注意されたい。また、展開のための離散化の粒度はあらかじめ固定されている。なお、プランニングコストとプランの質のトレードオフを考慮して、離散化の粒度を適応的に変化させることも可能である [三浦 98]。

以上の展開手法に基づいて、繰り返し改善を以下の手順で実行する。

(1): 現在有効なすべての候補を展開しその暫定コストを計算する。 $C_p^{temp}$  を候補  $p$  の暫定コストとする。

(2): 現在得られている最善の実行可能プラン  $fp^*$  のコスト  $C_{fp^*}$  を計算する ( $fp^* = \arg \min_p C_p^{temp}$ )。

(3) 各候補  $p$  の、次の展開後に得られるであろう新コスト  $C_p^{new}$  を見積もる。候補  $p$  のすべての開ノードをすべて展開したときに得られるコストの減少分 (プラン改善と呼ぶ)  $\Delta C_p$  が推定できるとすると\*<sup>3</sup>, 新コストは次式で与えられる:

$$C_p^{new} = C_p^{temp} - \Delta C_p. \quad (1)$$

この新コストが最良暫定解のコスト  $C_{fp^*}$  より大きい場合には、その候補は除外される。

(4) 更新された候補集合を基に次の繰り返しへ進む。

以上の繰り返しは任意の時点で終了することができ、繰り返しごとに  $C_{fp^*}$  が減少することが期待できるので、このプランニング過程は任意時間アルゴリズム (anytime algorithm) [Zilberstein 93] となる。なお、繰り返しの終了条件は次章で述べる。

## 5. スケジューリングアルゴリズム

提案するスケジューリング手法は、プランニングしながらの行動と行動しながらのプランニングを交互に実行する。これら2つの並列スケジューリングアルゴリズムは共にロボットが目的地に到着するまでの総時間の期待値を最小化することを目的とする。図6に並列実行の典型例をタイムチャートで示す。なお、初期FSSの計算は、行動しながらのプランニングによってすでに終了している場合には行わない。

### 5.1 プランニングしながらの行動の選択と実行

#### § 1 副目標候補集合と一貫性のある行動

有効な副目標 (次の観測位置) とは、その副目標を利

\*<sup>2</sup> この位置でロボットは不確かさなしでゲート幅を観測できると仮定する。

\*<sup>3</sup> ここでは、現在プランニングを行っている状況と似た状況に対するプランニングを過去に行っていることを仮定し、過去のプランニング事例からプラン改善を予測する手法 [三浦 98] を用いる。

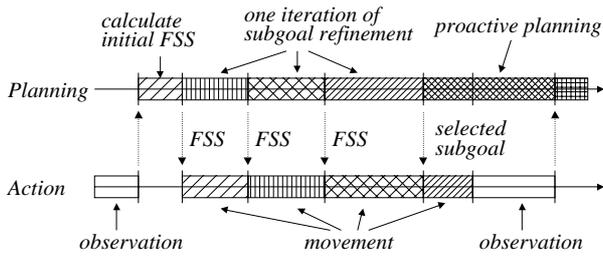


図6 並列実行のタイムチャート

用した場合の新コストの見積りが最良暫定コストより低いものである(4.2節参照)。したがって、有効な副目標集合(FSS)は、有効なプラン候補の最初の副目標(観測位置)の集合として生成される。繰り返し改善の1回の繰り返しごとに新たなFSSが計算される(図3参照)。

次に一貫性基準を定義する。次の繰り返しによってどの候補が除外されるかを事前に知ることはできないので、すべての副目標候補に対して一貫性のある行動を選択する。ある行動について、まず一般的な一貫性基準として

- すべての副目標までの距離を減少させると同時に、すべての副目標に対して、その行動よりよい行動の存在が自明でない。

という条件を用いる。さらに、問題固有の知識として、ロボットのジグザグ移動は好ましくないとして、

- その行動の終了地点から、すべての副目標に単一の円弧軌道で、後戻りすることなく到達可能であり、かつその円弧の半径がロボットの最小回転半径  $r_{min}$  以上である。

という条件も用いる。以上2つの条件を満たす行動の目標地点の集合が一貫性のある行動空間(CAS)を形成する。図7において、ロボットの現在位置からFSSへの2つの接線およびFSSで囲まれた領域が第1の基準を満たすものであり、その範囲内で第2の基準を満たす領域をCASとする。

なお、例えばFSSが空間的に大きな広がりを持つなどの理由で、一貫性のある行動がない場合には、次の繰り返し改善のステップを止まったまま行う。

§2 最適な行動の決定

本論文では簡単のため、行動としてFSS内の  $fp^*$  の位置へ向かう移動をとるものとする。実行コストを最小化するためには、CASの内部にある限り、できるだけ

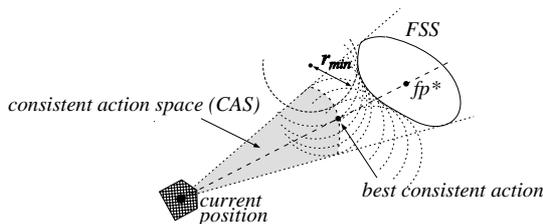


図7 最適な一貫性のある行動

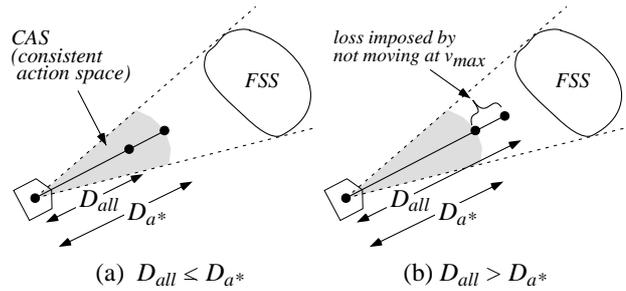


図8  $D_{all}$  と  $D_{a^*}$  の2つの可能な関係

け遠い地点を移動の目標位置とすればよい。そこで、行動の1つの候補は、現在位置から  $fp^*$  へ向かう直線とCASの境界との交点を目標位置とするものである。前項で述べた一貫性の2番目の基準から、この目標位置は直線とFSSに接する半径  $r_{min}$  の2つの円によってできる、2つの直線上の接点の、現在位置に近い方となる(図7参照)。

ここで次の2つの距離を考える。 $D_{a^*}$  は上記行動の移動距離である。 $D_{all}$  はロボットが繰り返し改善の次のステップの間、最高速度  $v_{max}$  で移動したときの移動距離である。

次の繰り返しに要する時間は次のように見積もる。各プラン候補  $p$  について、展開コスト(時間)  $C_p^{exp}$  を見積もる。これは、 $p$  のすべての開ノードの展開コスト\*4の和として計算できる。そうすると、次の繰り返しのコストは次式で与えられる:

$$C_{all}^{exp} = \sum_p C_p^{exp} \tag{2}$$

$D_{all}$  と  $D_{a^*}$  の関係が行動決定に重要な役割を果たす。以下に可能な状況を列挙し各状況に対する行動を考える。

(Case 1): もし  $D_{all} \leq D_{a^*}$  なら(図8(a)参照)、ロボットがCASから出るまでに次の繰り返し改善ステップを終えられるので、ロボットは全候補の繰り返し改善を行いながら最大速度  $v_{max}$  で移動する。1回の繰り返しが終了したら、更新されたFSSを基に新たな行動を決定する。

(Case 2): もし  $D_{all} > D_{a^*}$  なら(図8(b)参照)、現在の最良暫定解( $fp^*$ )を実行する代わりに、次の繰り返しステップを実行することは損失がある。すなわち、ロボットはCAS内にとどまるために、 $v_{max}$  よりも遅い速度で移動しなければならない。一方、 $fp^*$  を実行するときにはロボットは  $v_{max}$  で移動できる。前者の行動の損失は、距離  $D_{all} - D_{a^*}$  を速度  $v_{max}$  で移動する時間として計算できる。よって、2つのコスト、 $C_{fp^*}$  と候補の新コストの最小のものから上記損失を差し引いたものを、を比べ以下のように行動を決定する。

(Case 2-a): もし  $C_{p^*}^{new} + loss \leq C_{fp^*}$  なら、ロボッ

\*4 このコストはおよそ探索すべきノード数に比例した量である[三浦98]。

トは次の繰り返しステップを実行しながら、距離  $D_a$  だけ進む。繰り返しステップが終了したら、更新された FSS を基に次の行動を選択する。

(Case 2-b): もし  $C_{p^*}^{new} + loss > C_{fp^*}$  ならば、ロボットは候補の繰り返し改善を終了し、 $fp^*$  (その時点での最良解) を実行する。これが決定条件である。 $fp^*$  の実行時には、行動しながらのプランニングを行う (詳細アルゴリズムは次節)。観測後、ゲートの通過可能性が決定したら目的位置に向かって移動する。通過可能性が未定なら、プランニングしながらの行動の並列スケジューリングを再び行う。

以上の行動決定手続きの直観的な説明は次のようである。FSS は次に進むべき位置の候補集合を表しており、それが遠くにあるときには最高速度で進む。近くにあると最高速度で進むと FSS 内の一部の候補を通り越してしまうときには、速度を緩める。ただし、速度を緩めることによる損失が大きいときには、現在最良と考えられる候補を選択し実行する。

## 5.2 行動しながらのプランニング

### §1 現在の行動と一貫性のあるプランニングオプションの選択

3.2 で述べたように、現在の行動終了後に生じる可能性のある状態に対するプランニングオプションは、現在の行動と一貫性があるので、並列実行する価値がある。本論文で扱う問題では、ゲートの通過可能性が未定の場合に、さらなるプランニングが必要である。そこで、予測される未定状態 (図 9 の  $state_1 \dots state_n$  参照) に対するプランニングオプションは一貫性があると見なす。未定ノードは固定化した粒度で離散化するので (4.2 参照)、あらかじめ可能な未定状態を数えあげることができることに注意されたい。

### §2 最適なプランニングオプション集合の決定

行動しながらのプランニングに利用できる時間  $T_{pwa}$  は、ある副目標が選択されてから、それが達成されるまでの時間、すなわち、副目標を選択した時点での位置から観測位置までの移動時間と観測時間の和で計算できる。

1 つのプランニングオプションは、対応する未定状態に対するプラン候補を 1 レベルだけ改善する過程とする。そこで、時間  $T_{pwa}$  内で実行可能かつプラン改善の期待効用を最大化するようなプランニングオプションの組を選ぶ。効用として、プランニングオプションをあらかじめ実行することにより節約できる時間を用いるが、これは各

プランニングオプションを実行するために必要な時間そのものである。

以上の方針で、次のようにプランニングオプション集合を決定する。まず各未定状態に対し初期候補集合 (FSS) を生成し、各オプションの実行時間を見積もる。ここで、 $i$  番目の状態について、 $P_i$  をその状態が生じる確率、 $t_i$  をその状態に対するオプションの実行時間とする。 $t_i$  は式 (2) の  $C_{all}^{exp}$  に等しい。 $i$  番目のプランニングオプションの期待効用は  $P_i t_i$  で与えられる。

ここで、最初に与えられた時間  $T_{pwa}$  から初期候補の集合に要した時間を差し引いたものを  $T'_{pwa}$  とする。もし、プランニングオプションの実行時間の和 ( $\sum_{i=1}^n t_i$ ) が  $T'_{pwa}$  より小さければ、すべてを実行する。そうでなければ次の最適化問題を解く:

- 目的関数:  $\sum_{i=1}^n P_i t_i w_i \rightarrow \text{最大}$ ,
- 拘束条件:  $\sum_{i=1}^n t_i w_i \leq T'_{pwa}$ ,  $w_i = 0 \text{ or } 1$ .

ここで  $w_i$  は  $i$  番目のプランニングオプションを実行するかどうかを示すフラグである。これは 0-1 ナップサック問題である [茨木 83]。

すべてのプランニングオプションを実行した後にまだ時間が残っていたら、次のレベルの改善へ進む。各レベルごとに、上に述べたような、実行すべきプランニングオプション集合の決定を行う。

## 6. シミュレーション

図 10 と図 11 にシミュレーションに用いた例題と、ゲートの初期通過可能確率を変えて行った 2 つのシミュレーション結果を示す。ロボットは、ゲートが通過可能であればゲートへ真っ直ぐ向かい、不可能であれば迂回路への入り口へ真っ直ぐ向かう。したがって、次の観測位置 (副目標) は、図中中央の三角形内に限定できるので、その内部にグリッドを設定し、観測位置と移動位置の候補をグリッド上に限定した。三角形以外の線分はゲートあるいは迂回路を通して目的地へ至る経路を示す。初期位置から目的地までのゲートを通った最短距離は約 680[cm]、最高移動速度 ( $v_{max}$ ) は 5[cm/s]、1 回の観測時間は 4[s] とした。また開ノードの離散化の粒度 (4.1 節参照) は 5 とした。

図 10 は初期確率 0.4 の場合である。通過可能性が決定するまでにロボットが要した時間は 119.7[s] である。プランニングと行動を逐次的に実行した場合に比べ、プランニングしながらの行動、行動しながらのプランニングそれぞれにより、17.23[s] (12.6%)、3.55[s] (2.9%) が削減できた。図 11 は初期確率 0.7 の場合である。実行時間は 93.0[s]、削減された時間の割合は 12.9% と 1.3% であった。目的地までの期待到達時間最小という規範でプランニングするため、初期通過確率が高い場合には三角形領域の左上辺により近い位置で観測し、低い場合には下辺により近い位置で観測することが期待される。図中の

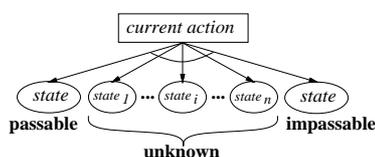
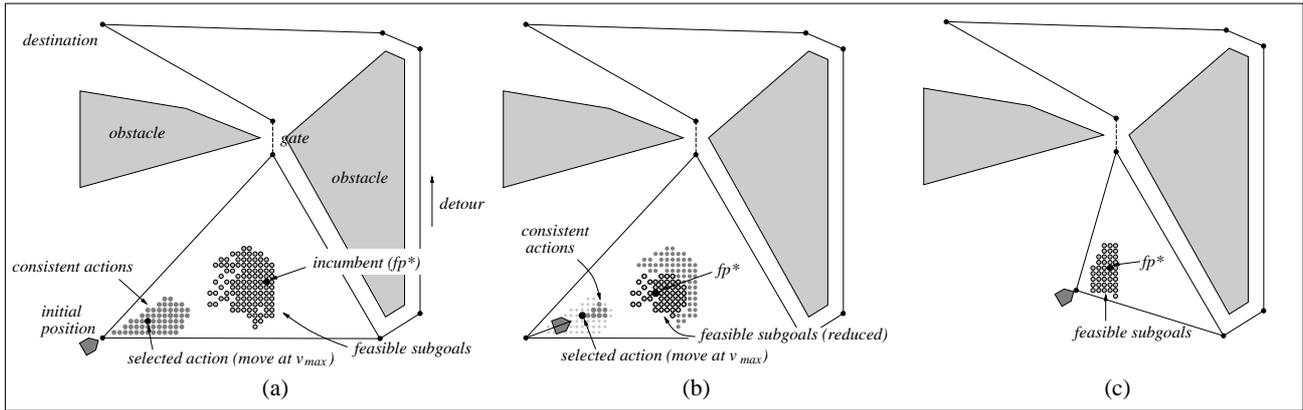
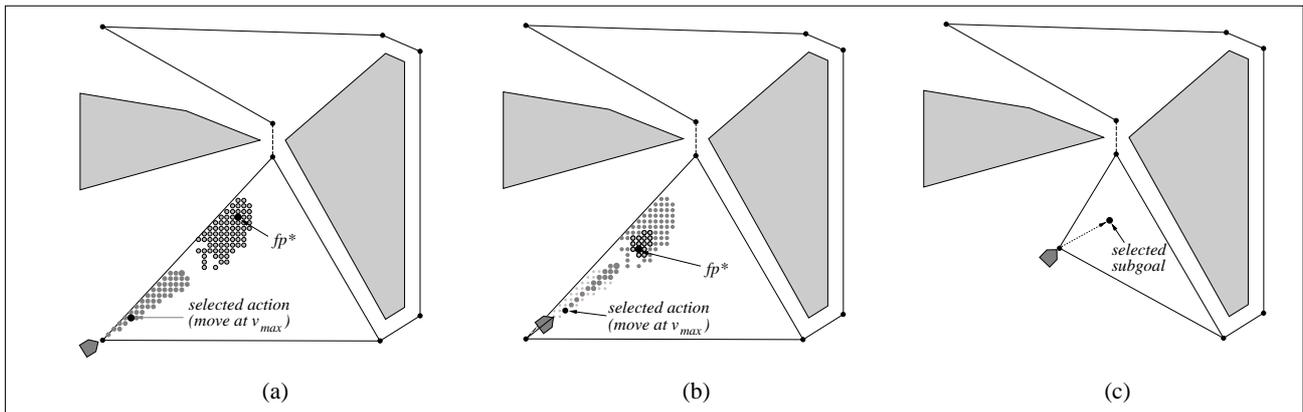


図 9 予測される状態



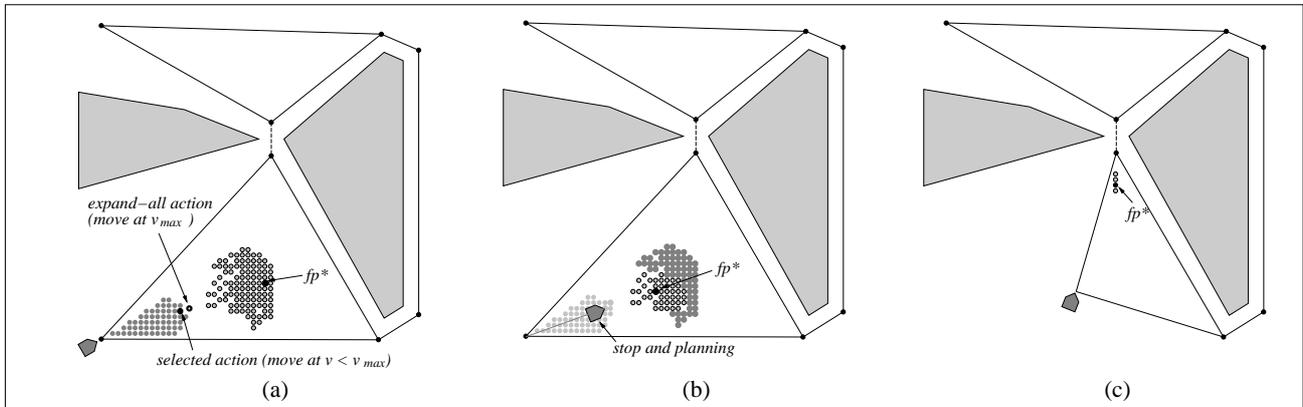
(a) 初期の FSS, CAS から最大速度  $v_{max}$  での移動を選択し実行した. 移動と並列して FSS 内の各候補を精密化した. (b) 更新された FSS と CAS から再び  $v_{max}$  での移動を選択し, 精密化しながら移動した. 移動後  $fp^*$  で示される候補を選択した. その候補の実行と並行して, 予測された 17 の観測後の状態のうち 7 の状態に対するプランニングオプションを実行した. (c) 観測後の実際の状態に対するプランニングオプションをすでに実行していたが, それでもなお FSS が広がり過ぎて CAS が得られなかったため, 停止して候補の精密化を実行した. 1 回の精密化の後,  $fp^*$  を選択し実行した.

図 10 シミュレーション結果 1



(a) 初期の FSS, CAS から最大速度  $v_{max}$  での移動を選択し実行した. (b) 2 回の精密化の繰り返しの後,  $fp^*$  を選択し実行した. その実行中に, すべての予測される状態に対するプランニングオプションを実行した. (c) すべての状態に対する解がすでに得られていたので, ロボットはすぐに次の観測位置へ移動した.

図 11 シミュレーション結果 2



(a) 3.1 § 2 の (Case 2-a) の条件が成り立ったので,  $v_{max}$  の約 92% の速度で CAS 内に留まるように移動する行動を選択した. (b) CAS が得られなかったため, 停止して候補の精密化を行い, その結果  $fp^*$  を選択した. (c) 観測後の FSS を基に  $v_{max}$  で移動しながら候補の精密化を行い,  $fp^*$  を選択した.

図 12 シミュレーション結果 3

表 1 ゲートの初期通過確率と並列実行による利得の関係

initial prob.	by awp (%)	by pwa (%)	(maximum)
0.1	13.46	0.01	(1.07)
0.3	14.81	0.16	(3.00)
0.5	16.01	0.16	(3.29)
0.7	16.48	0.20	(3.04)
0.9	10.87	0.15	(2.58)

awp: プラニングしながらの行動

pwa: 行動しながらのプラニング

FSS の出現位置にそのような傾向が現れている。

表 1 は並列スケジューリングによる時間削減の平均 (100 回の試行) に対する, 初期確率の影響を示したものである。ここで, 行動しながらのプラニングによる削減が非常に少ないのは, いくつもの予想した未定状態のうち, たった 1 つだけが実際に起こるためである。そのために, 最大の時間削減も合わせて示してある。並列スケジューリングによる利得 (時間削減) は基本的に調べるべき候補の数に依存する。したがって, 状況がより不確実であればあるほど (初期確率が中くらいであるほど), 並列スケジューリングはより有効である。表 1 からそのような傾向が見てとれる。

図 12 では図 10 の設定から, ロボットの最大移動速度を 2 倍にするという変更だけを加えた場合のシミュレーション結果である。この場合, CAS 内にとどまるためには最大速度で移動することはできない。3.1 § 2 で述べた, 速度を落とすことによる損失を考慮した行動比較により, 結果として速度を落として移動しながら, 現在の FSS 内の候補集合を精密化するスケジュールを選択した。

## 7. おわりに

本論文では, 行動しながらのプラニングに加え, プラニングしながらの行動も可能にする, プラニングと行動の並列スケジューリング手法を提案した。プラニング (行動) と並列に実行可能な行動 (プラニング) の集合を決定するために, プラニングと行動の一貫性という考え方をういた。視覚情報の不確かさの下での移動ロボットの視覚誘導問題に対するスケジューリングアルゴリズムを開発した。シミュレーション実験によって, 提案する手法の有効性を示した。現在, 本手法を実ロボットに実装して評価することを行っている。

本論文では, ロボットの視覚誘導問題を対象としたが, プラニング部と行動部が並列に実行可能なシステムであれば, 本論文で提案する手法を他のエージェントのスケジューリング問題に適用することができる。プラニングしながらの行動に関しては, 3.1 で述べたように, 副目標候補を繰り返的に絞っていくプランナを構築すると共に, 2 つの基準 (一貫性基準と決定基準) を設定する必要がある。特に, 一貫性基準は並列度に大きな影響があると思われるので, 問題ごとに適切なヒューリス

ティックを選択することが重要である。また, 行動しながらのプラニングについては, 観測を含む行動の結果を (少なくとも確率的に) 予測できることが必要である。他のスケジューリング問題に対する提案する手法の有効性の検討も今後の課題である。

## ◇ 参考文献 ◇

- [Boddy 89] Boddy, M. and Dean, T.: Solving Time-Dependent Planning Problems, in *Proceedings of IJCAI-89*, pp. 979-984 (1989).
- [Horvitz 97] Horvitz, E.: Models of Continual Computation, in *Proceedings of AAAI-97* (1997).
- [茨木 83] 茨木俊秀: 組合せ最適化 - 分枝限定法を中心として, 産業図書 (1983).
- [Kambhampati 95] Kambhampati, S., Knoblock, C., and Yang, Q.: Planning as Refinement Search: A Unified Framework for Evaluating Design Tradeoffs in Partial-Order Planning, *Artificial Intelligence*, Vol. 76, pp. 167-238 (1995).
- [Korf 90] Korf, R.: Real-Time Heuristic Search, *Artificial Intelligence*, Vol. 42, pp. 189-211 (1990).
- [Miura 97] Miura, J. and Shirai, Y.: Vision and Motion Planning for a Mobile Robot under Uncertainty, *Int. J. of Robotics Research*, Vol. 16, No. 6, pp. 806-825 (1997).
- [三浦 98] 三浦, 白井: プラニングコストと視覚情報の不確かさを考慮した移動ロボットの視覚と行動のプラニング, *人工知能学会誌*, Vol. 13, No. 4, pp. 588-596 (1998).
- [Nourbakhsh 97] Nourbakhsh, I.: *Interleaving Planning and Execution for Autonomous Robots*, Kluwer Academic Publishers (1997).
- [Russell 91] Russell, S. and Wefald, E.: *Do The Right Thing*, The MIT Press (1991).
- [Zilberstein 93] Zilberstein, S.: *Operational Rationality through Compilation of Anytime Algorithm*, PhD thesis, University of California at Berkeley (1993).

[担当委員: × × ]

19YY 年 MM 月 DD 日 受理

## 謝 辞

本研究の一部は文部省科学研究費の援助のもとに行われた。

## 著 者 紹 介

三浦 純 (正会員)

1984 年東京大学工学部機械工学科卒業。1989 年同大学院工学系研究科情報工学専攻博士課程修了, 工学博士。同年大阪大学助手。現在同大学院工学研究科電子制御機械工学専攻助教授。知能ロボット, 人工知能, コンピュータビジョンの研究に従事。1994 年 ~ 1995 年, CMU 客員研究員。1997 年日本ロボット学会論文賞受賞。日本ロボット学会, 電子情報通信学会, 情報処理学会, 日本機械学会, IEEE, AAAI 各会員。

白井 良明 (正会員)

1964 年名古屋大学工学部機械工学科卒業。1969 年東京大学大学院工学系博士課程修了, 工学博士。同年, 電子技術総合研究所入所。1985 年制御部部長。コンピュータビジョン, ロボティクスの研究に従事。1971 年 ~ 1972 年, MIT AI ラボ客員研究員。1988 年大阪大学工学部電子制御機械工学科教授。