

# 状況的行為に基づいた近似計画手法

## Post-hoc Reconstruction of Plans by Situated Actions

イ セリン\*<sup>1</sup> 久保田 孝\*<sup>2</sup> 中谷 一郎\*<sup>3</sup>  
Serin Lee Takashi Kubota Ichiro Nakatani

\*<sup>1</sup>東京大学大学院 The University of Tokyo  
\*<sup>2,3</sup>宇宙航空研究開発機構・宇宙科学研究本部 JAXA-Institute of Space and Astronautical Science

The approaches to make an agent generate intelligent actions in the AI field might be very roughly categorized into two ways—the classical planning and situated action system. However, each system actually has its own application field as well as its strength and weakness, and, in particular, the application of the situated action system is usually restricted to the field such as the navigation of mobile robots which is not directly related to the logical problem. This paper proposes a novel action generator to situatedly extract a set of actions from the relaxed logical space, which is likely to help to achieve the goal at the current situation. After performing the action set, the agent should recognize the situation for deciding the next likely action set. The empirical result in some planning domains shows that the quality of the resultant path to the goal, *post-hoc reconstruction of plans*, is mostly acceptable as well as deriving the fast response time. Therefore, this approach could make the agent such as mobile robots and web agents in the dynamic environment respond to the environment fast.

### 1. はじめに

エージェントは問題解決、又は目的達成のため自ら適切な行動を生み出すことが要求される。このような行動の発生的手法を提案したり、又は生物でのそのような現象を説明することを目的に多くの研究分野が発達してきた。特に人工知能 (AI) 分野では大きく分けて次の二つのアプローチで研究が進んできた。

プランニング 目的達成までの全過程を決める手法であり、状況と行動に関する推論から目的までの一貫した行動パターン、即ち計画を立てる。

状況的行為 (Situated action) プランニングの抽象的な推論に対する反発として提案されたとみなすことができ、現状だけを考慮して行動を生み出す [Lucy 87]。

プランニング手法は現状態から目的を満たす状態までの経路をエージェントの行動に関する論理的な推論から求める。一般的にそのような推論は膨大な情報とリソースを必要とするため、計画を立てるためには比較的長い時間を必要とする。従って、動的な環境では状態の変化に追いつかない場合が生じる。

一方、状況的行為に基づいた手法は抽象的な推論より現状の考察を重視するため、一般的に行動を引き出す時間が速い。従って、未知の動的な環境のように将来の予測が難しい場合には、プランニングより高い状況性を期待できるが、目的までの一貫した行動のパターンを引き出すことは難しい。また、実際にはこのような概念を応用する際の様々な難しさのため、プランニング手法が対象としている論理的な問題を扱えない。従って、このようなアプローチは主に論理的な問題から幾何学的な問題に置き換えられる移動ロボットのナビゲーションに適用されている [Arkin 98][Brooks 86]。

本研究では、動的な環境で複雑な論理的問題を解くことができるエージェントの開発を目的としている。そのため、上記の二つのアプローチを次のように兼ね合わせた手法を提案する。

連絡先: イ セリン, 宇宙科学研究本部中谷研, 〒 201-0015 神奈川県相模原市由野台 3-1-1, tel: 042-759-8311, e-mail: srlee@nsl.isas.ac.jp

現状で実行できる行動から、目的の達成に要求されると見なされる行動のみを抽出して、それを実行するという過程を繰り返せば最終的に目的が達成できる。

引き続き、第 2 章では最近のプランニング手法について簡単に紹介し、第 3 章に上述の提案について詳しく述べる。第 4 章では、提案されたアプローチの実験結果を示す。

### 2. プランニング

#### 2.1 Notation

状態  $S$  は論理的な要素 (事実) の有限な集合から構成されている。プランニング問題は  $P = (O, I, G)$  である。ここで、 $O$  はエージェントのアクションに関する記述、 $I$  は世界の初期状態に関する記述、 $G$  はエージェントに与えた目的に関する記述を示す。アクション  $o \in O$  は STRIPS アクションであり、 $o = \{pre(o), add(o), del(o)\}$  で定義される。ここで、 $pre(o)$  は行動  $o$  の前提条件、 $add(o)$  は  $o$  が状態に加える要素、 $del(o)$  は状態から消去される要素を示す。一つの STRIPS アクションの結果は前提条件が満たされれば、即ち  $pre(o) \subseteq S$  の場合に  $Result(S, < o >) = (S \cup add(o)) \setminus del(o)$  になる。前提条件が満たされない場合は定義されない。

現時点で実行できる行動によって現状態に加えられる要素の中、目的の達成に要求されると見なされる事実の集合を  $\mathcal{F}_{ag}$  と示す。ここで、 $\mathcal{F}_{ag}$  を形成する行動の集合  $\mathcal{A}_{ag}$  は現時点で同時に実行可能な行動だけを含んでいるものではない。即ち、互いに排他的な行動も含む可能性があるため、 $\mathcal{A}_{ag}$  の中で矛盾がなく同時に実行できる行動のみの集合を別に  $\mathcal{L} \subseteq \mathcal{A}_{ag}$  と定義する。即ち、エージェントは毎状況から  $\mathcal{L}$  を抽出し、 $\mathcal{L}$  の要素を実行することを繰り返せば最終的に目的の状態に到達できると期待される。

#### 2.2 探索による計画手法

1969 年 Green の手法以来様々な計画アルゴリズムが提案されてきた。その中で、最近のほとんどの計画手法は  $O, I, G$  から作られたプランニング空間で探索を行うことによって計画を引き出すアプローチを取っている。このような計画手法の中で Graphplan の例を図.1 に示す [Weld 99][Blum 97]。

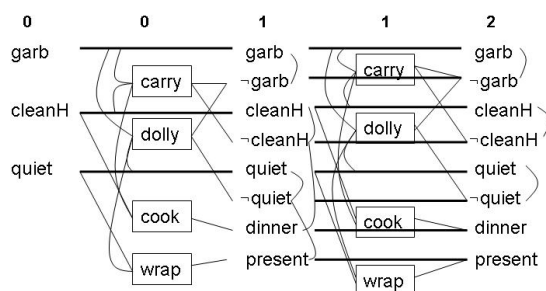


Figure 1: dinner date 問題のプランニンググラフ: プランニンググラフとは有向グラフで、一つのレベルは状態を示す集合とその状態から実行できるアクションの集合から構成されている。従って、最初のレベルの状態は現状態に、最後のレベルは目標が達成できたと見られる状態に対応している。細い線は互いに排他的な関係 (*mutex*; mutually exclusion relation) を示している。以前のレベルの状態の要素は次の状態レベルに *noop*(*maintain action*) を通じて伝播される。プランニンググラフの作成は最終のレベルに  $\mathcal{G}$  が非排他的に含まれた場合完了し、そのレベルからレベル 0 に向かって探索を行いながらプランを抽出する。もし、その探索の中に排他的な要素が含まれている場合、前のあるレベルに戻って再び探索を行う。

最近には Bonet によって提案されたヒューリスティック探索によるアプローチ、即ち近似化されたプランニング空間からヒューリスティックを求めて、元の空間で高速に探索を行う手法が非常に優れたパフォーマンスを見せている [Bonet 01][Hoffman 01][Gerevini 02][Vidal 04].

### 3. 状況的行為の導出

#### 3.1 近似化された論理空間

論理空間から  $\mathcal{L}$  を抽出するために本稿では次のような近似化を提案する。

目的達成のため、ある状況の要素をいつ実行すべきかは考慮しない。ただ、現在達成できる事実の中で、目的に到達するために必要な事実はどのようなものなのかだけを考慮する。

このような近似化では、現状態の後のある時点で実行できる行動によって加えられる事実の優先を求めていない。そのため、そのような行動の消去効果、即ち状態から消去する部分は考慮しない<sup>\*1</sup>。また、目的達成のため必要な事実をいつ実行するかを考慮しないため、一つの行動をどのようなタイミングで実行すべきかについても考慮しない。従って、行動は任意のある時点で一回だけ考慮する。本稿では、ある時点の状態から実行できるアクションをその時点の後に伸ばして実行する場合を考慮しない。

$\mathcal{F}_i$  と  $\mathcal{A}_i$  をそれぞれプランニンググラフの事実レベル  $i$  とアクションレベル  $i$  で定義すると、上記の近似化は次のプランニング問題に関するグラフ空間で記述できる。

- $\mathcal{F}_0, \mathcal{A}_0, \mathcal{F}_1$  に対して、 $\mathcal{P} = (\mathcal{O}, \mathcal{I}, \mathcal{G})$

\*1 状態から消去される部分のため、行動の間や事実の間には様々な排他的な関係が生じる。例えば、ある行動の前提条件が他の行動の効果によって消去されると、二つの行動は現時点で排他的な関係になる。

- 他のレベルに対しては、 $\mathcal{P}' = (\mathcal{O}', \mathcal{F}'_1, \mathcal{G})$ 。ここで、 $\mathcal{O}' = \{(pre(o), add(o), \emptyset) \mid (pre(o), add(o), del(o)) \in \mathcal{O}\}$ ,  $\mathcal{F}'_1 = \mathcal{F}_1 \setminus del(o_i) (\forall o_i \in \mathcal{A}_0)$

- ただ、 $o \in \text{level } i$  なら、 $o$  は  $\text{level } j > i$  には登場しない。

- 最後の状態レベルは  $\mathcal{G}$  の全ての要素 (事実) を含んでいる。

#### 3.2 $\mathcal{L}$ の抽出

近似的な論理空間  $\mathcal{R}$  を作った後、最後の状態レベルに含まれている  $\mathcal{G}$  の要素からレベル 0 に向かって、状態とそれを達成する行動、またその行動の前提条件を抽出する。前節で述べたように、アクションの実行を後に延ばすことは考慮しないため、ある事実を達成するアクションを抽出する時にはできる限り *noop* を選ぶ。もし、ある事実を達成する *noop* が存在しない場合には *non noop* からランダムに選べる。

このような *noop first heuristic with randomly selected non noop* によって抽出されたレベル 1 の事実の集合が  $\mathcal{F}_{ag}$ 、それを達成するレベル 0 の行動の集合が  $\mathcal{A}_{ag}$  になる。

$\mathcal{L}$  は  $\mathcal{A}_{ag}$  の中から、他の行動を妨害しないものを選ぶことによって求められる。もし、そのような行動が存在しない場合には  $\mathcal{A}_{ag}$  または、 $\mathcal{A}_0$  からランダムに行動を選べる<sup>\*2</sup>。

しかし、このように  $\mathcal{L}$  を  $\mathcal{A}_{ag}$  や  $\mathcal{A}_0$  からランダムに選ぶことによって、エージェントは論理的空間での行き止まり (*dead end*) に収束することがある。本稿ではこのような行き止まりを含んでいる問題を考慮しない。また、 $\mathcal{G}$  まで短い (効率的な) 経路を取らない場合も生じる。特に与えられた  $\mathcal{G}$  自体に順番が存在する場合には非常に長い経路を取る可能性がある。しかし、近似的に目的の順番を分析し、プランナに並べ換えられた  $\mathcal{G}$  を与えられる手法によって、この問題は多少解ける [Koehler 00].

### 4. 経験的な結果

本稿では既存のプランニング問題に対して、提案したアプローチと他の二つの STRIPS プランナ—FF v2.3 と LPG-td(speed) との優劣を比較するため実験を行った。提案したアプローチと二つのプランナは C で作成され、他の特殊なライブラリは使用していない。

しかし、提案したアプローチと他のプランナの目的が違うため、公平な比較の基準を採ることは難しい。

本稿では、提案したアプローチが目的達成まで  $\mathcal{L}$  を抽出する時にかかった時間、即ち、反応時間の平均と他のプランナが全手順を再プランニングを行わず決めた時間を比較した。また、提案したアプローチによる事後解析的なプランの長さ、他のプランナが出したプランの長さも比較した<sup>\*3</sup>。

実験は、プランニングの研究で昔から使われてきた domain と最近のプランニング大会で使われた domain を合わせて、四つの domain—Block's domain, Logistics domain, Satellite domain, Rovers domain で行われた。このような domain は論理的空間で行き止まりを含んでいない。また、Block's domain は  $\mathcal{G}$  自体に順番が存在するため、この場合には Koehler の GAM (Goal Agenda Management) を、提案したアプローチに適用した [Koehler 00].

\*2 実際には  $\mathcal{A}_{ag}$  ではなく、 $\mathcal{A}_0$  から  $\mathcal{L}$  を求めなければならない場合はあまり生じない。そのため、本稿では  $\mathcal{L}$  を率  $\zeta = 0.05$  で  $\mathcal{A}_0$  から求める。

\*3 Schuman は計画を状況的行為の post-hoc reconstruction とし て考えた [Lucy 87]. 本稿でもそのような立場を取っている。

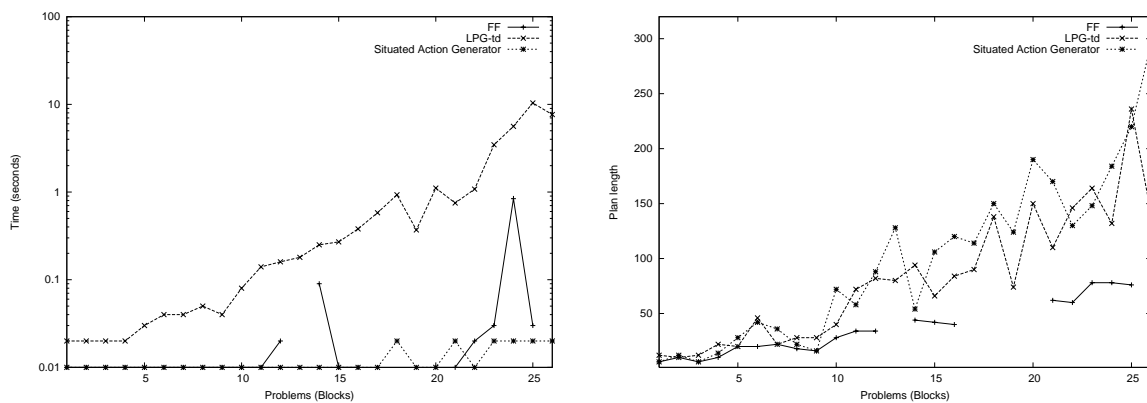


Figure 2: 時間とプランの長さ (Blocks problems)

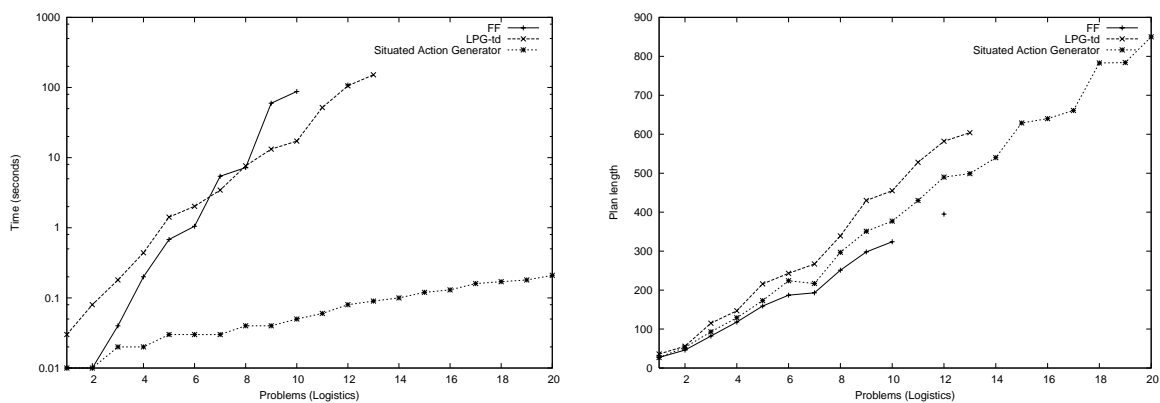


Figure 3: 時間とプランの長さ (Logistics problems)

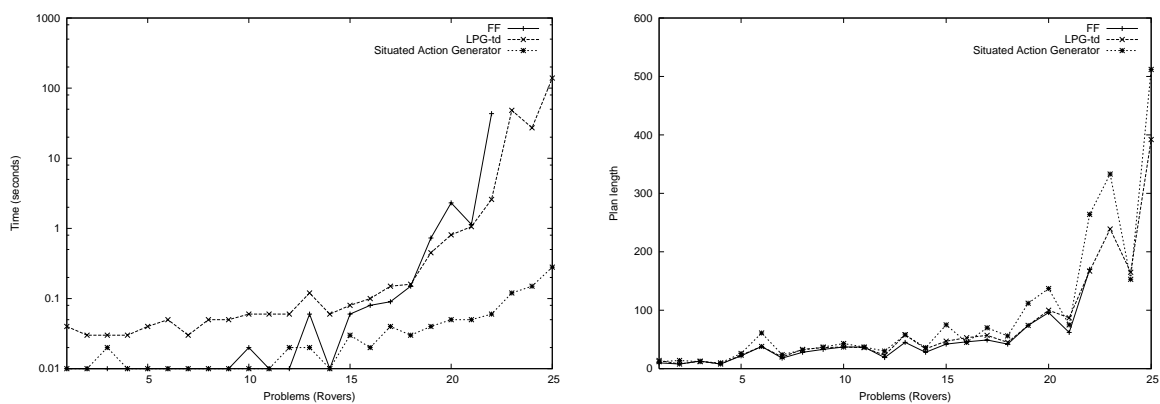


Figure 4: 時間とプランの長さ (Rovers problems)

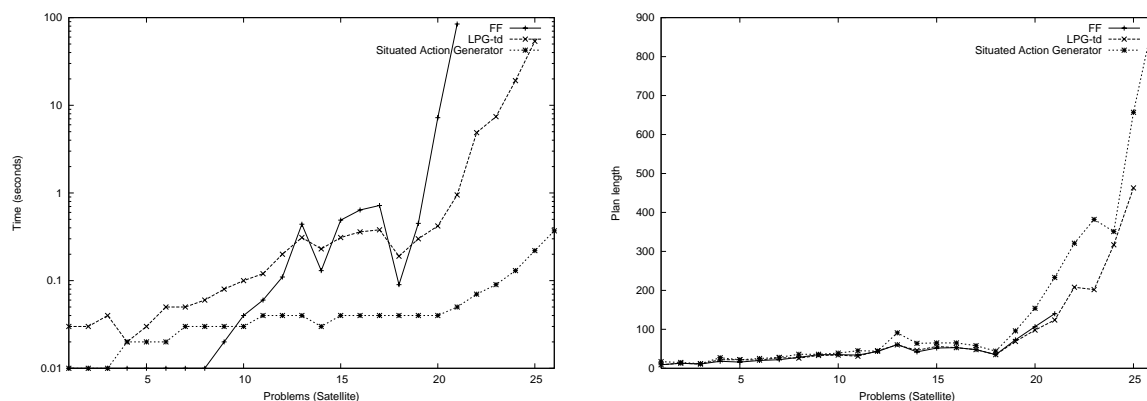


Figure 5: 時間とプランの長さ (Satellite problems)

全ての実験は Intel Pentium4 の 1.7GHz, 768MB の Linux 2.6 で行った。また、制限時間は 200 秒に設定した。結果を図.2 から図.5 に示す。

小さい問題に対しては、状況的行為による事後解析的なプランの長さは、既存のプランナが出したプランの長さと比較してもあまり差が出なかった。

大きい問題に対してはその差が少し大きくなった。しかし、提案したアプローチは使用するリソースが比較的少ないため、既存のプランナが解けない問題を解くことができた。例えば、ある Rover 問題に対しては、提案したアプローチが平均的に総メモリの 30% を使って問題を解いたが、他のプランナは 70% 以上のメモリを使っても問題を解けなかった。

## 5. おわりに

本稿では論理空間で状況的行為を引き出す手法を提案した。提案したアプローチは、目的達成のためある事実をいつ実行すべきかは考慮せず、現在達成できる事実の中で、目的に到達するために必要な事実はどのようなものなのかだけを考慮することによって状況的行為を引き出すものである。

引き出された状況的行為を実行した後は、また状態を認識して状況的行為を引き出す過程を繰り返せば、問題が行き止まりを含んでいない限り、最終的に目的が達成できる。

様々な domain での実験結果は、少ないリソースを使用しながらも、環境の変化に瞬時に応じられることを示している。また、状況的行為による事後解析的なプランの長さは、既存のプランナが出したプランの長さと比較しても受け入れられる程度であることがわかった。

また、提案したアプローチは、エージェントが動的な環境で複雑な作業を行う場合に適切であると思われるが、特に目的までの期待される経路が短く、状況自体が非常に複雑な場合に、優れたパフォーマンスを示すと期待される。我々の毎日の生活もこのような状況に近いであろう。

行き止まりを避けるために既存のプランナと共存する知能構造の開発などが今後の課題である。

## References

- [Lucy 87] Lucy Suchman, Plans and Situated Actions - The Problem of Human-Machine Communication, Cambridge University Press, 1987
- [Arkin 98] Ronald C. Arkin, Behavior-Based Robotics, The MIT Press, 1998
- [Brooks 86] Rodney A. Brooks, A robust layered control system for a mobile robot, *IEEE Journal of Robotics and Automation*, April, 1986
- [Blum 97] Avrim L. Blum, and Merrick L. Furst, Fast Planning Through Planning Graph Analysis, *Artificial Intelligence*, 1997
- [Bonet 01] Blai Bonet, and Héctor Geffner, Planning as Heuristic Search, *Artificial Intelligence vol. 129 (1-2)*, 2001
- [Hoffman 01] Jörg Hoffman, and Bernard Nebel, The FF Planning Systems: Fast Plan Generation Through Heuristic Search, *Journal of Artificial Intelligence Research* 14 253-302, 2001
- [Gerevini 02] Alfonso Gerevini, Alessandro Saetti, and Ivan Serina, Planning through Stochastic Local Search and Temporal Action Graphs in LPG, *Journal of Artificial Intelligence Research*, 2002
- [Vidal 04] Vincent Vidal, A Lookahead Strategy for Heuristic Search Planning, *AAAI-04*, 2004
- [Weld 99] Daniel S. Weld, Recent Advances in AI Planning, *AI Magazine*, 1999
- [Nabeshima 01] Hidetomo Nabeshima, and Katsumi Inoue, Planning Graph and SAT Planning, *Journal of the Japanese Society for Artificial Intelligence*, vol. 19, no. 3, 2001
- [Koehler 00] Jana Koehler and Jörg Hoffman, On Reasonable and Forced Goal Orderings and their Use in an Agenda-Driven Planning Algorithm, *Journal of Artificial Intelligence Research*, vol. 12, 2000