

古典的プランニング問題に対するプラン最適化アルゴリズムとその組み合わせ

Combining Plan Optimization Algorithms for Classical Planning Problems

遠藤翔馬^{*1} 浅井政太郎^{*1} 福永 Alex^{*1}
Shoma Endo Masataro Asai Alex Fukunaga

^{*1}東京大学大学院総合文化研究科
Graduate School of Arts and Sciences, The University of Tokyo

The quality of plans generated by domain-independent planners can be improved by applying plan optimization algorithms which perform local optimizations. We propose a system which iteratively applies multiple optimization algorithms. We show that a combination of optimization passes can improve quality further than the application of a single optimization algorithm.

1. はじめに

人工知能分野におけるプランニング問題とは、初期状態から望ましい状態に移行するために必要な行動の順序、すなわちプランを求める問題である。プランニング問題では解となるプランのコストが重要なテーマであり、同じ問題を解決するプランであればより小さいコストで目的を達せられている解の方を「質が良い」とみなす。古典的プランニング問題の一般的な定式化である STRIPS モデルにおいては、最適解、つまり最小のコストでその問題を解決するプランを探索する問題を解くことは非常に難しく、その問題クラスは NP 困難に属することが知られている [Bylander 94]。こうした問題の特徴から、プランニング分野では最適解を求めることは難しくてもできるだけ質の良い解を少ないリソースで効率良く探索する解法が多数提案されてきた (代表的な性能を持つものとして [Helmert 06] [Richter 10] 等)。

プランの探索過程を改良することで得られる解の質を高めようとする研究の一方で、そうした探索の結果得られた解を後から改善する方法も提案されている。すなわち既に得られている解をさらに入力として受け取り、後処理を施して、改善された解を出力する、という手法である (図 1, 2)。本稿ではこうした手法を総称してプラン最適化 **plan optimization** と呼ぶ。近年の研究では、プラン生成を行うプランナを単独で運用するより、プラン最適化アルゴリズムと併用した方がより効率良く質の高いプランを得られたとする実験結果が報告される [Nakhost 10] など、プラン最適化の有用性が示され始めている。

本稿では先行研究で提案されているアルゴリズムをより効率良く適用するための手法である randomized window 法を新規に提案する。そしてこれらのアルゴリズムの効果を実験によって確かめる。また個々のアルゴリズムの性能を個別に検証するだけでなく、それぞれの手法を組み合わせることで、より良い結果が得られることを示す。



図 1: 一般的なプラン生成アルゴリズムの流れ

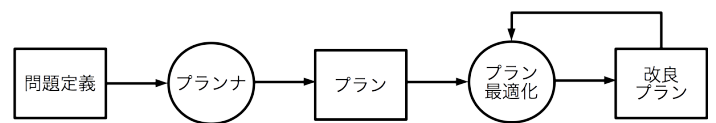


図 2: プラン最適化アルゴリズムの流れ

2. プラン最適化の先行研究

本節では既に提案されているプラン最適化の手法、アルゴリズムをいくつか紹介する。

2.1 Action Elimination

Action Elimination (AE) はプラン中に存在する不要なアクションを排除することによってよりコストの小さい部分プランを得る手法である [Nakhost 10]。Nakhost らは近似的に質の良い部分プランを探索する貪欲アルゴリズムを提案している。このアルゴリズムは入力プラン中の各アクションを二重ループで点検していき、もしそのアクションが行われなかったと仮定しても問題なくプランが成り立つかどうかを確認するもので、入力プラン π の長さを n 、ひとつのアクションが持つ前提条件の最大数を p としたとき、時間計算量は $O(n^2 \times p)$ となる。

2.2 Action Dependency Analysis

Chrpa らはプラン中に含まれるアクション同士の相互の依存関係を解析することで、不要なアクションを発見する手法を提案した [Chrpa 12]。

アクション間の依存関係について、Chrpa らはプランの中で先に現れるあるアクションが後に現れるあるアクションの前提条件を提供しているような関係を依存と定義している。定式的にはアクションの列 (a_1, \dots, a_n) において $i < j$ のとき、 $(\text{eff}^+(a_i) \cap \text{pre}(a_j)) \neq \emptyset$ かつ $(\text{eff}^+(a_i) \cap \text{pre}(a_j)) \not\subseteq \bigcup_{t=i+1}^{j-1} \text{eff}^+(a_t)$ であるようなアクション a_i と a_j の関係を直接的な依存とし、この直接的依存関係が推移的に成り立っているアクション同士の関係を依存と定義した。ここで $\text{eff}^+(a)$ 、 $\text{pre}(a)$ はそれぞれアクション a の追加効果、前提条件を表す。Chrpa らは前提条件がゴール条件と等しく、追加効果も削除効果も持たない特殊なアクション a_g をプランの最後尾に追加したとき、 a_g と依存の関係にないアクションは取り除くことができることを証明した。

さらにこの依存関係解析を利用することで、逆向き **inverse** のアクションのペアを発見することができる。逆向きのアクションとは、例えば「A 地点から B 地点へ向かう行動」と「B 地点から A 地点へ向かう行動」のように、互いに効果を打ち消し合うようなアクションのことである。この例の場合、もしこの 2 つのアクションが立て続けに行われていけば、それは明らかに冗長な行動である。Chrpa らは依存関係解析を行うことで、こうした打ち消し合って結果的に無意味となるアクションのペアを発見する方法についても提案している。

Chrpa らの提案するアルゴリズムの計算量はプランの長さを n 、「逆向きのアクションのペア」の数を k 、「逆向きのアクションのペア」に挟まれている他のアクションの最大数を l としたとき、 $O(n^2 + kl)$ となる。

2.3 Plan Neighborhood Graph Search

Plan Neighborhood Graph Search (PNGS) は入力プランをグラフで表現したとき、その近傍に存在する経路を探索しよりコストの小さいプランを得るアルゴリズムである [Nakhost 10]。より形式的には、入力プランを基に、元々のプランニング問題の探索空間の極めて小さな部分グラフを生成し、その中で最短経路を計算するものである。

プランニング問題そのものは、グラフの探索問題として考えることができる。すなわち状態がノードであり、アクションがエッジの有向グラフである。このような表現においてプランニング問題は「探索空間全体を表す巨大なグラフから初期状態のノードからゴール条件を満たすノードへの経路を探索する」問題に帰着される。PNGS ではこうした問題全体のグラフを考えるのではなく、まず入力プランに現れる状態とアクションだけを含む、一本道のグラフを作成する。次にグラフのそれぞれのノードについて L 個ずつ新たなノードとエッジを付け足してグラフを拡張する。この手順によって得られるグラフは、問題全体を表すグラフの非常に小さな部分グラフになっている。そしてこのグラフから改めて最短経路を探し、得られた経路 (= プラン) を現在のプランと置き換える。この一連の流れを L の値を増やしつつ反復し、リソースの尽きた時点で得られている最新のプランを出力とする。

2.4 その他の手法

紙幅の都合上詳細は割愛するが、他にも本探索時の各ノードにおけるヒューリスティック値の動きから有望なショートカットが存在しそうな区間を重点的に探索し直す AIRS アルゴリズム [Estrem 12] や、プランをブロックと呼ぶグループに分解してからブロック単位で最適化を行う block decomposition アルゴリズム [Siddiqui 12, Siddiqui 13] などが提案されている。

2.5 複数の最適化手法の組み合わせ

本節で紹介した先行研究のプラン最適化手法は、相互に排他的なものではない。プラン最適化は入力も出力もプランであるため、プランナが生成したオリジナルのプランを改善するだけでなく、他の最適化アルゴリズムが出力したプランを再び入力に取って別の最適化アルゴリズムにかけるということも可能である。実際、AE と PNGS についてはどちらか一方だけを用いるよりも両方を組み合わせる方が全体的には良い結果が得られるという実験結果が示されている [Nakhost 10]。

3. 提案手法

以上で見たように既にいくつかのプラン最適化アルゴリズムが提案されているが、本研究ではこうした手法を非常に大規模な問題に適用するための新たな枠組みを提案する。それはプ

ランの一部区間をランダムに取り出し、その区間のみで最適化を行う手法である。

先行研究のアルゴリズムの問題点のひとつは、長いプランを扱うことが難しいという点にある。紹介したアルゴリズムはいつでもプラン全体を走査して最適化するという性質上、計算量がプランの長さに強く依存する。特に AE やアクション依存解析はプラン長の 2 乗が計算量に係ってくることから、プランが長くなれば長くなるほど最適化に多くの時間がかかる。

こうした問題点を解決するために本研究では窓 **window** の概念を導入した。窓とは、関数の一部の区間を切り取る窓関数のように、プランの一部の区間を切り取る概念である。窓によってごく短い一部を切り出しその区間のみを最適化の対象とすることで、極端に長いプランでも効率良く最適化を行うことができると考えられる。窓をどのように決めるかについては様々な方法を考えることができるが、本研究では長さも位置もランダムに決める randomized window を採用した。アルゴリズムは、入力に最適化対象のプラン、用いる最適化アルゴリズム、許容する最大のウィンドウの長さ、時間制限を取り、何度もウィンドウを変えながらプランの一部を切り出し別の最適化アルゴリズムに渡す、という一連の流れを繰り返し、時間制限に達した際の最新のプランを出力する。

4. 性能評価

先行研究のアルゴリズムと提案手法を実装し、ベンチマーク問題集を解くことでこれらのアルゴリズムが大規模な問題に対してプラン最適化の効果をもちうるかを検証した。

計算機実験の実行環境には Intel Xeon E5410@2.33GHz を用いた。個々のアルゴリズム単位での並列化は行わず、1 つの問題について 1 コアを使って解いた。

ベンチマーク問題には工場でのセル生産方式の大量製品組立をモデル化した assembly-mixed ドメインを用いた。最適化の対象とするプランについては、CAP プランナの出力した解を使用した (ドメイン、プランナ共に [Asai 15])。

実験の対象としたアルゴリズムは、2 節で紹介した action elimination(AE)、action dependency analysis(ACTDEP) と、提案手法である randomized window の最適化処理部分に PNGS を採用したものの (WPNGS) である。先行研究のアルゴリズムについては、論文で公開されているものを独自に実装したものを使用した。またいずれのアルゴリズムも、時間制限を最大 10 分間に設定した。

実験では CAP プランナの出力した assembly-mixed ドメインの問題のプランに対して各アルゴリズムによる最適化を行い、CAP オリジナルのプランと比較してどの程度コストの削減されたプランが得られるかを検証した。結果を表 1 に示す。

実験結果から、個々のアルゴリズム単体での最適化効果が確認できるだけでなく、それらを組み合わせることで適用していったときにさらなる改善が見られることがわかる。また本研究において提案した randomized window も確かに最適化効果をもたらしていることが確認できた。

最も改善効果が高かったのは ACTDEP、WPNGS、AE の順に適用したものであり、全体的に最初に ACTDEP を適用したときの群が良い結果を示している。これは ACTDEP が比較的軽い処理であり、最長のプランに対しても制限時間の 10 分以内にすべての処理を完了できていたことが要因のひとつと考えられる。AE は長いプランだと時間内に全ての区間を改善しきれないケースがあるため、先に ACTDEP で処理されて短くなったプランに後から AE をかけた結果高効率な改善がな

表 1: assembly-mixed ドメインの問題を CAP で解いたプランに最適化を行った結果. コストは全プランの値を合計したあと, CAP オリジナルの値が 100 になるよう正規化した. 複数の手法が+で連なっているものは, 左の手法から順に適用していったことを表す.

algorithm	cost
CAP	100.00
AE	94.92
AE + ACTDEP	89.43
AE + WPNGS	94.02
AE + ACTDEP + WPNGS	88.43
AE + WPNGS + ACTDEP	89.31
ACTDEP	91.90
ACTDEP + AE	84.78
ACTDEP + WPNGS	90.57
ACTDEP + AE + WPNGS	84.22
ACTDEP + WPNGS + AE	83.84
WPNGS	98.41
WPNGS + AE	93.91
WPNGS + ACTDEP	90.64
WPNGS + AE + ACTDEP	88.47
WPNGS + ACTDEP + AE	88.38

れたのではないかと推測される.

5. 関連研究

[Asai 15] は [Botea 04] で提案されたコンポーネント抽象化 **component abstraction** の概念の発展形やそれに基づいたマクロアクション生成手法などを利用し, 大規模問題を分割して解く手法を提案している. 本研究の提案手法では window をランダムに選択して最適化を行ったが, これらの概念に基いて window の選択を行うことで, プランの「自然な区切り」に相当するような箇所を検出することが可能になるのではないかと考えている. このような window の設定戦略は今後探求すべき課題のひとつである.

またプラン最適化は, コンパイラ最適化との類似性がある. プランが行動の列であるように, プログラムは命令の列である. そしてコンパイラ最適化が入力プログラムより実行効率の良いプログラムを導出することを目的としているように, プラン最適化は入力プランよりコストの小さいプランを求めることを目的としている. こうした類似性から, コンパイラ最適化の知見を取り入れることがプラン最適化の研究に良い影響を与えることが期待できる. 特に複数の最適化アルゴリズムの効率のよい組み合わせ方や適用順序の決定法については, コンパイラ最適化分野において既に多くの研究が行われている ([Almagor 04][Cavazos 07] 等). この問題はプラン最適化においても重要な課題であり, コンパイラ最適化分野の研究成果の応用を検討することには意義があるだろう.

6. まとめ

本研究では既存研究で提案された三つのプラン最適化法を組み合わせることにより, 個々の手法より質の良い解を得ることが可能であることを実証した.

今後の課題としては, より効果的な新たな手法の開発や既存のアルゴリズムの実装面からの性能改善などが挙げられる. 特

に今回の実験においては先行研究のアルゴリズムを独自に実装したため, 元のアルゴリズムが外部のプランナと連携して効率的な計算を行っている点などを再現しきれずに性能が劣化した可能性が懸念される. 既存の最適化アルゴリズムの組み合わせ戦略を考える上でもアルゴリズムの性能の再現性は重要な要素であるため, この点は早急に是正すべき課題である.

参考文献

- [Almagor 04] Almagor, L., Cooper, K. D., Grosul, A., Harvey, T. J., Reeves, S. W., Subramanian, D., Torczon, L., and Waterman, T.: Finding effective compilation sequences, *ACM SIGPLAN Notices*, Vol. 39, No. 7, pp. 231–239 (2004)
- [Asai 15] Asai, M. and Fukunaga, A.: Solving Large Scale Planning Problems with Component Macros, in *ICAPS* (2015)
- [Botea 04] Botea, A., Müller, M., and Schaeffer, J.: Using Component Abstraction for Automatic Generation of Macro-Actions., in *ICAPS*, pp. 181–190 (2004)
- [Bylander 94] Bylander, T.: The computational complexity of propositional STRIPS planning, *Artificial Intelligence*, Vol. 69, No. 1, pp. 165–204 (1994)
- [Cavazos 07] Cavazos, J., Fursin, G., Agakov, F., Bonilla, E., O’Boyle, M. F., and Temam, O.: Rapidly selecting good compiler optimizations using performance counters, in *Code Generation and Optimization, 2007. CGO’07. International Symposium on*, pp. 185–197 IEEE (2007)
- [Chrapa 12] Chrapa, L., McCluskey, T. L., and Osborne, H.: Optimizing Plans through Analysis of Action Dependencies and Independencies., in *ICAPS* (2012)
- [Estrem 12] Estrem, S. J. and Krebsbach, K. D.: AIRS: Anytime Iterative Refinement of a Solution., in *FLAIRS Conference* (2012)
- [Helmert 06] Helmert, M.: The Fast Downward Planning System., *J. Artif. Intell. Res. (JAIR)*, Vol. 26, pp. 191–246 (2006)
- [Nakhost 10] Nakhost, H. and Müller, M.: Action Elimination and Plan Neighborhood Graph Search: Two Algorithms for Plan Improvement., in *ICAPS*, pp. 121–128 (2010)
- [Richter 10] Richter, S. and Westphal, M.: The LAMA planner: Guiding cost-based anytime planning with landmarks, *Journal of Artificial Intelligence Research*, Vol. 39, No. 1, pp. 127–177 (2010)
- [Siddiqui 12] Siddiqui, F. H. and Haslum, P.: Block-structured plan deordering, in *AI 2012: Advances in Artificial Intelligence*, pp. 803–814, Springer (2012)
- [Siddiqui 13] Siddiqui, F. H. and Haslum, P.: Plan quality optimisation via block decomposition, in *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pp. 2387–2393 AAAI Press (2013)