

トランスポジションテーブルを利用した IDA*探索の閾値による並列化

Parallel-Window IDA* with a Transposition Table

中野雄基^{1*} 福永 アレックス¹
Nakano Yuki¹ Fukunaga Alex¹

¹ 東京大学大学院総合文化研究科

¹ Graduate School of Arts and Sciences, The University of Tokyo

Abstract: We investigate a parallel iterative-deepening A* algorithm where each iteration of IDA* is executed on a separate thread, and a shared transposition table is used in order to minimize redundant search.

1 はじめに

本研究では、トランスポジションテーブル (TT) を利用した IDA*探索を提案して実験的に検証する。

ヒューリスティック探索である A*探索は、初期状態からノード n にたどり着くまでのコストを $g(n)$ 、ノード n から目標状態までのヒューリスティック値を $h(n)$ として、 $f(n) = g(n) + h(n)$ を計算して、未展開のノードの内、 f が最初のノードを展開しながら探索を行う。また、同種の一つである IDA*探索 [Korf 85] は反復深化深さ優先探索を行う探索アルゴリズムである。IDA*探索ではノードの f 値を反復深化の閾値で制限し、閾値を増加させながら再探索を行い、解を発見するまで探索を繰り返す。IDA*探索は A*探索を深さ優先探索に変換したアルゴリズムであるため、A*探索に比べメモリ使用量が遥かに少ない特徴がある。しかし、展開したノードの情報をメモリに記録しないため、一度展開したノードを重複して展開してしまう欠点がある。

TT とは探索が済んだ状態を保存するハッシュ表である。本研究ではノードのハッシュ値をキーとして、ノードの「 g 値、 h 値、展開したプロセスの閾値、初期状態からの経路」を保存している。規模の大きな問題では TT にすべての状態を保存することはできないため、置き換え規則が必要である。

Reinfeld [Reinfeld 94] らは IDA*探索において TT を用いることでノードの再展開を制御する手法を提案した。赤木ら [Akagi 10] は TT に記録する情報により、アルゴリズムの完全性及び最適性が損なわれる可能性を指摘して、完全性及び最適性を保証する手法を提案した。

IDA*探索では閾値を増加させながら再探索を繰り返す。純粋な IDA*探索では、再探索の際にそれ以前の探索の結果を利用しないため、ある閾値ごとに行われる探索は独立した挙動をする。そのため、Powley らは異なる閾値ごとにプロセスを立ち上げ、アルゴリズムの並列化を行った [Powley 91]。

2 提案手法

本研究では Parallel Window IDA*において一つの共有 TT を用いたアルゴリズムを提案し、その性能を検証した。

・IDA*の並列化

IDA*探索の反復深化では A*探索における f 値によって、探索空間を制限している。本研究では、異なる閾値を並列のプロセスに割り当てることで並列化を実現した。探索済みの閾値のリストは全プロセスが共有しており、同じ閾値を他のプロセスが計算することは無い。

一方、並列 IDA*の場合、最適解の f 値より大きい閾値で探索を行うプロセスが発生してしまうことがある。従って、最初に解を発見したプロセスが最適化解を発見できる閾値を探索しているとは限らないため、発見された解の最適性の保証を行う必要がある。これに対する対処法として、閾値 t で探索中のプロセスが解を発見した際、閾値が t より大きいプロセスは直ちに探索を終了して、 t 未満の閾値で探索中のプロセスは引き続き探索するを継続する。探索を継続したプロセスの中で最も閾値の小さいプロセスが解を発見した際にはそれを最適化解として出力する。これ以外のプロセスが解を発見した際にはより小さい閾値で探索しているプロセスの終了を待つ。

*東京大学大学院総合文化研究科広域科学専攻広域システム科学系
〒153-8902 東京都目黒区駒場 3-8-1 15 号館 504B
E-mail: nakano.yuk1.hs@gmail.com

プロセスの初期化, 閾値の割当 (N 並列)

1. (初期化)1~N プロセスに1~N の閾値を与え, これらの値を探索済みとする
2. 与えられた閾値で IDA*探索を終了したプロセスは, 閾値により枝刈りされた状態の中で最も小さい評価値 (f 値) を次の「閾値の候補」とする
3. 「閾値の候補」が未探索であれば, この閾値で探索を開始する
4. 「閾値の候補」がすでに探索済みであれば, 未探索の閾値の中で最小のものを次の閾値候補とする

・並列 IDA*への TT の利用

ヒューリスティック探索ではノードのヒューリスティック値の計算が実行時間の大部分を占めるため, ノードのヒューリスティック値を TT を保存することで探索の高速化をはかる. TT を利用する際には, 書き込みと読み込みを別々に行われており, 読み込みは, ノード n の展開 (ヒューリスティック値を計算) 前にテーブルにノード n が記録されているか確認を行い, テーブルにあれば登録されたヒューリスティック値を利用する. また, テーブルに登録されている g 値がノード n の g 値より小さい場合はノード n は冗長であるため展開を中止する. 書き込みは, テーブルにノード n が登録されていなかった際, ヒューリスティック値を計算後にテーブルへの書き込みを行う. 並列探索ではこの際のロックが必要になり, これがオーバーヘッドの原因になる.

探索では TT に保存された情報から, 状態 n の展開が不要であることが証明されれば, n 以下状態の探索を枝刈りできる. しかし, TT の容量は有限であるため, 効率的な枝刈りのためには頻出する状態を優先的に TT に記録する必要がある. 赤木らの研究では, 4 手法でテーブルに置き換え規則を検証していたが, 本研究では以下の方法を提案する. TT は複数のプロセスから同時にアクセスされるため書き込み, 読み込みの際にロックが必要である. 本研究では次のように書き込みを行い, プロセス数を増加させた際に発生するテーブルへの書き込みのロック待ちを緩和させるとともに, 探索において頻出する状態 (ノード) は確率的に書き込み回数を増加させた.

TT への書き込み

他のプロセスがテーブルをロック中
TT に書き込みは行わずに, 探索を継続

テーブルのロックを獲得
テーブルに現在展開中のノードの情報を書き込む

3 評価と考察

ドメイン非依存プランナの Fast Downward [Helmert 06] において提案手法を適用して, 並列プランナを実装した. 並列化には MPI を用いた. プランニングの分野で標準ベンチマークとして使用されている IPC (international planning competition) 問題集より 107 問を選び, 性能評価を行った. 一問あたり実験時間は 30 分, メモリは 2GB に制限した. 提案手法を 4 コアで実行して, 比較対象として 1 コアで赤木らのアルゴリズム [2] を実行した. 提案手法は基本的には赤木らのアルゴリズムを並列化したものであるため, 提案手法を N コアで実行した場合, 最大 N 倍の高速化が期待できる.

ドメイン (#問題数)	提案手法	TT を用いた IDA*
airport(#50)	25	20
blocks(#35)	19	17
depot(#22)	2	2
total(#107)	46	39

上の表は 107 問中, 時間内に解けた問題の数を示している. 提案手法が赤木らの逐次アルゴリズムより多くの問題が解けている. 探索時間についても両アルゴリズムが解けた問題に限定して比較した結果, 提案手法が赤木らの逐次手法と比べて 2 - 3 倍の高速化を得られた. 以上の結果から, トランスポジションテーブルを用いた IDA*探索の並列化による高速化はある程度実現されたと考えられる. 本研究では, TT を利用した IDA*探索を並列化を試みた. 今後の課題として, 最もキャッシュヒット率の高いテーブルの置き換え規則の検証, 探索の閾値の振り分け方の検証, 分散環境における性能評価, などがあげられる.

参考文献

[Korf 85] Korf R.: Depth-First Iterative- Deepening: An Optimal Admissible Tree Search: *Artificial Intelligence*, Vol. 25, pages 97-109, 1985.

[Powley 91] Powley C, Korf R.: Single-agent parallel window search. *IEEE Transactions on Pattern Analysis and Machine Intelligence: International Journal of Examples*, Vol.13, No.5, pages 466-477,1991.

[Akagi 10] Akagi Y, Kishimoto A, Fukunaga A: On transposition tables for single-agent search and planning: Summary of results. *Third Annual Symposium on Combinatorial Search* 2010.

[Helmert 06] M. Helmert : The Fast Downward Planning System *Artif. Intell. Res.(JAIR)* Vol.26,pages191-246,2006.

[Reinefeld 94] Reinefeld, A., and Marsland, T. : Enhanced iterativedeepening search. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol16,pages 701-710.