研 究 報 告 書

「大規模並列化によるハイパフォーマンス人工知能システム」

研究期間: 平成20年10月～平成24年3月
研 究 者: 岸本 章宏

1，研究のねらい

本研究では、大多数の計算機を利用して、人工知能技術の超高速化の研究を遂行する。本研究の最終目標は、数千 CPU コアを利用した計算機環境でもアルゴリズムの高速化が達成できる並列化手法を開発することである。並列化の対象には、人工知能分野の基盤技術である探索アルゴリズムを主に利用する。多くのアプリケーションで用いられる探索アルゴリズムは、組み合わせ爆発の生じる大規模な空間を高速に探索し、実時間で解を求める必要があるので、大規模並列化による高速化は、アプリケーションの性能改善のための重要なアプローチである。また、本研究は、探索空間を大規模な情報とし、解を有益な情報とみなせば、本領域の「大規模な情報から知識を生産・活用するための基盤技術の創出」という目標に合致する。

2，研究成果

本研究の主な成果は次の通りである。

（1）プランニングにおける大規模並列探索アルゴリズムの開発

プランニング・システム(プランナー)では、ユーザーが解きたい問題を PDDL や STRIPS などのプランニング言語で記述すれば、ユーザー自身が解法を考えなくてもプランナーが自動的に解を求めてくれるので、ユーザーにとっては様々なドメインに簡単に適用できるシステムである。このため、汎用プランナーの産業界からの需要は、将来高まっていくことが予想される。

本研究では、高性能なプランナーであり、研究者間で最も広く用いられている Fast Downward を並列化した。具体的には、Fast Downward では、人工知能分野の基盤アルゴリズムである A*アルゴリズムが用いられており、この A*アルゴリズムの大規模分散並列版である HDA* (Hash Distributed A*)を開発した。

分散並列 A*の問題点は、どのようにして効率よくプロセッサ間で探索結果を保持し、その探索結果を再利用することで無駄な並列探索を省略できるかにある。逐次 A*では、探索結果の参照の際にローカルのメモリをアクセスすればよいだけであり、この問題自体が生じないのに対し、分散並列 A*では、ある計算ノードが他の計算ノード上にある探索結果にアクセスするためには、通信や同期オーバーヘッドが生じる。これが分散並列 A*の性能低下の原因である。

本研究の HDA*では、データ駆動型スケジューリングという考え方を提案し、データ（ハッシュ表やテーブルのエントリー）のあるところに必ず仕事（計算）を移動させることによって、探索結果の効率的な再利用を実現しつつ、非同期並列計算を駆使できるようにした。その結果として、HDA*は、並列アルゴリズムの性能を低下させる大きな原因である通信遅延を

隠ぺいできるようになり、高い高速化率を達成できるようになった。

HDA*の性能評価は、プランニング分野で最もスタンダードな問題集を解かせることによって行った。その結果、HDA*は 1024 コアで最大 600 倍の高速化を達成しただけでなく、分散環境の莫大なメモリ(1-2TB)を利用することによって、これまでの高性能なプランナーで解けなかった問題のいくつかを高速に解くことに初めて成功した。

（2）大規模並列ゲーム木探索アルゴリズムの開発

ゲームは、社会で利用されている実アプリケーションに比べ、ルールと結果が単純であるのにもかかわらず、計算量の組み合わせ爆発が生じるので、難しい問題である。このため、人工知能分野における理想的な題材として、約60年にわたって研究されてきた。

UCT アルゴリズムは、最近特にコンピュータ囲碁の分野で注目されているゲーム木探索アルゴリズムである。ルート並列化は、UCT アルゴリズムの並列化法であり、簡単でかつ広く用いられている。

本研究では、まずルート並列化を改良(合議制ルート並列化と呼ぶ)し、ルート並列化との性能を囲碁プログラムで比較した。性能評価実験では、両手法を取り入れた囲碁プログラムで、64CPU コアを利用して、プログラムの強さを調べた。合議制 Root 並列化は、Root 並列化よりも性能が良かったが、CPU コア数を増加させれば、両手法には大きなボトルネックがあることが分かった。 このため、合議制ルート並列化やルート並列化では、非常に限定的な性能向上しか達成できず、64 コア以上の CPU を利用しても性能が改善しないことが分かった(64 コアでも 2-3 倍程度の高速化)。性能が向上しない原因としては、UCT の構築する探索木が大きければアルゴリズムの性能が向上するのにもかかわらず、ルート並列化では、この探索木をコア間で共有しないために、ルート並列化が構築する探索木の実質的な大きさは、逐次 UCT とほぼ同じであることが挙げられる。

本研究では、以上の通り、ルート並列化の限界を示した後、プランナーの大規模分散並列化でも利用した考え方である、データ駆動型スケジューリングを駆使し、探索木を CPU コア間で効率良く共有する並列 UCT アルゴリズムを開発した。この新しい並列 UCT では、データ駆動型スケジューリングの特長である計算の非同期化と、ルート並列化では行えなかった、メモリの効率的な利用による探索木のサイズの巨大化が実現できたのであるが、特定の CPU コアが過剰のメッセージを受け取ってしまうという問題が生じた。そこで、UCT と同等の性能を持つ深さ優先探索アルゴリズム DFUCT を並列化の対象に変更し、データ駆動型スケジューリングに基づいて、大規模分散並列化を行った。この大規模分散並列 DFUCT では、スタンダードなテスト問題である pgame において、4800CPU コアで最大で約 1800 倍という高い台数効果が得られた。

（3）逐次探索アルゴリズムの改良

本研究に着手してから、A*アルゴリズムを代表とする多くの逐次探索アルゴリズムでは、そもそもメモリを大量に利用するため、大規模並列化によってメモリサイズを大幅に増やしたとしても、メモリ不足のせいで解を求められないことが多いことが判明した。このため、先行研究のアルゴリズムよりもメモリの利用量が大幅に少ない逐次探索アルゴリ

ズムをいくつか開発し、プランナーやゲーム・プログラムの能力を向上させることに成功した。これらの新しい探索アルゴリズムの並列化は、研究成果(1)(2)で駆使したデータ駆動型スケジューリングを利用すれば、効率良く行えると期待できる。
さらに、15 年以上前に発表され、基礎技術として広く用いられている探索アルゴリズム EIDA* (Enhanced IDA*)アルゴリズムに間違いがあることを発見した。本研究では、この EIDA*の誤りを正しく修正し、理論的に常に正しい結果を返せることを保証した。

（4）Binary Decision Diagram (BDD)構築の並列化

データ駆動型スケジューリングを用いた分散並列化の3つ目の対象アルゴリズムとして、BDD 構築を選んだ。BDD は、二値論理関数のコンパクトな表現方法であり、ハードウェア回路検証やデータ・マイニング、機械学習やパズルなど、様々な応用分野がある。
本研究では、北海道大学の湊真一研究室で開発されている BDD パッケージの逐次ソース・プログラムを入手し、並列化を行った。テストドメインには、N-Queen 問題の解を列挙する問題を利用した。本研究で行った並列化のアプローチは次の二つである。
まず、BDD 構築アルゴリズムの性質を調べるため、8 コアを持つ共有メモリ計算機上でアルゴリズムの並列化を行った。この並列化では、3.5 倍程度の高速化を達成できた。
次に、データ駆動型スケジューリングを駆使し、分散環境で BDD 構築の大規模並列化を行ったが、600 コアを利用した場合に約 20 倍程度遅くなってしまった。この性能低下は、BDD 構築アルゴリズムがデータ駆動型スケジューリングを利用したとしても、ゲームやプランニングよりも通信遅延を隠ぺいしにくい手法であることに起因した。この問題の解決策は、今後の課題にしたい。

3, 今後の展開

本研究で成功を収めたアルゴリズムである HDA*や大規模分散並列 DFUCT は、当然プランニングやゲームにそのまま利用できる。しかし、本研究で開発したアルゴリズムで共通して利用する並列化法のアイデア（つまり、データ駆動型スケジューリング）をどの程度まで一般化できるかを考えれば、他のアルゴリズムの並列化にも(例えば、分散並列ダイナミックプログラミング)利用できる可能性が生じるので、本研究の適用範囲が自然に広がる。

HDA*と大規模分散並列 DFUCT では、データ駆動型スケジューリングを利用する点が同じであるのにもかかわらず、現状では、対象とするアプリケーションを並列化する際には、それぞれのアプリケーションに合わせて、データ駆動型スケジューリングのソースコードをスクラッチから記述している。

データ駆動型スケジューリングを一般的に適用できるようにするためには、このような職人芸的な開発スタイルから脱却し、他の研究者やプログラマーがもっと手軽にデータ駆動型スケジューリングに基づく並列プログラムを開発できる環境を整備しなければならない。例えば、並列ライブラリや並列プログラミング言語などを設計・公開することによって、大規模分散並列アルゴリズムの開発を支援できれば理想的である。今後の研究の展開としては、プログラミング言語研究者と協力しながら、データ駆動型スケジューリングに介在する基本性質を解析し、その知見に基づいて、データ駆動型スケジューリング用の基本プリミティブを設計することが挙げられる。

4，自己評価

本研究の当初の予定は、探索アルゴリズムの代表的なアプリケーションであるゲームとプランニングを研究題材として、数千コアを用いた大規模並列化を行うことであった。実際に、我が国を代表とするスーパーコンピュータである東京工業大学の TSUBAME 上で1000コア以上用いた実験を行えたこともあり、当初の計画よりも早いペースでこれらの分野で効率の良いアルゴリズムの開発を進めることができただけでなく、予想以上の大きな研究成果を出すことができた。特に、プランニングでの成果について執筆した論文は、当該分野の最高峰の国際会議 ICAPS'09 で最優秀論文賞を受賞（ICAPS で日本人が最優秀論文賞を受賞するのはおそらく初めての快挙である）し、本研究のインパクトの高さを国際的に示せたと考えている。

当初の計画よりも研究が順調に進んだため、さきがけ開始当初には考えていなかった研究である、逐次探索アルゴリズムの性能改良と BDD 構築の並列化を行った。逐次探索アルゴリズムについては、15年以上にわたり利用されているアルゴリズムの理論的な欠陥を修正したり、プランナーやゲーム・プログラムの能力を大幅に向上させたりするなど、重要な成果をあげることができた。開発したこれらのアルゴリズムは、ドイツやカナダの大学院のセミナー等でも取り上げられ、これらのアルゴリズム改良に関する研究が国際的に進み始めている。BDD 構築の並列化に関しては、8コア程度の共有メモリ型計算機上ではある程度の成功を収めたものの、数百コアを利用する大規模分散並列 BDD 構築アルゴリズムの開発には成功しなかった。この BDD 構築の分散並列化に関しては、新たな手法を考えて、再挑戦したい。

人工知能分野での代表的なアルゴリズム開発・改良では成功を収めたものの、本研究領域での目標の一つである情報社会への応用については、社会で実際に利用されているシステムに組み込むという段階までには、残念ながら到達できなかった。この原因の一つとして、最も適用範囲の広いアルゴリズムである BDD 構築の大規模並列化が成功しなかったことが挙げられる。しかし、今回の研究で駆使した考え方である「データ駆動型スケジューリング」自体は一般的な考え方であり、このスケジューリングに基づく並列プログラミング言語や並列ライブラリを開発できれば、さきがけで得られた知見が社会で実際に利用される機会は自然と広がると考えられる。この点については、5-10 年にわたる長期的な計画であるが、今後の大きなテーマにしていきたい。

5，研究総括の見解

囲碁等のゲームを素材とした大規模探索というテーマである。探索は大規模データからの知識獲得のための基本方式であるため、今後、学習や知識発見との関わりを付けて、知識創出に関係する研究の方向が出てくることを期待していた。

当初の計画よりも早いペースで、ゲームとプランニングの分野で、効率の良い大規模並列化アルゴリズムの開発を進めることができ、予想以上の大きな研究成果を出すことができている。また、当初の計画よりも研究が順調に進んだため、計画にはなかった逐次探索アルゴリズムの性能改良と BDD 構築の並列化も行っており、高く評価したい。

一方、アルゴリズム開発・改良では成果を上げているが、情報社会への応用については十分な成果を得られていない。今回の研究で駆使した考え方である「データ駆動型スケジューリング」に基づく並列プログラミング言語や並列ライブラリを開発できれば社会で実際に利用される機会

が広がると考えられ、今後のテーマにしていってほしい。

6, 主な研究成果リスト

(1)論文(原著論文)発表

1. Akihiro Kishimoto, Alex Fukunaga, and Adi Botea. "Scalable, Parallel Best-First Search for Optimal Sequential Planning", In Proceedings of the 19th International Conference on Automated Planning and Scheduling (ICAPS'09), pages 201-208, 2009.
2. Yusuke Soejima, Akihiro Kishimoto and Osamu Watanabe. "Evaluating Root Parallelization in Go", IEEE Transactions on Computational Intelligence and AI in Games, Volume 2, Number 4, pages 278-287, 2010
3. Yuima Akagi, Akihiro Kishimoto and Alex Fukunaga. On Transposition Tables for Single-Agent Search and Planning: Summary of Results. In Proceedings of the 3rd Symposium on Combinatorial Search (SoCS'10), pages 2-9, 2010,
4. Tatsuya Imai and Akihiro Kishimoto. "A Novel Technique for Avoiding Plateaus of Greedy Best-First Search in Satisficing Planning", In Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI-11), pages 985-991, 2011
5. Kazuki Yoshizoe, Akihiro Kishimoto, Tomoyuki Kaneko, Haruhiro Yoshimoto and Yutaka Ishikawa. "Scalable Distributed Monte-Carlo Tree Search", In Proceedings of the 4th Symposium on Combinatorial Search (SoCS'11), pages 180-187, 2011

(2)特許出願

研究期間累積件数:0 件

(3)その他の成果(主要な学会発表、受賞、著作物等)

招待講演

1. 岸本章宏, Alex Fukunaga, Adi Botea.「最適解を求めるプランニング・アルゴリズムの大規模並列化について」, 第22回回路と軽井沢システムワークショップ, 2009.

著書

1. 小谷善行、岸本章宏、柴原一友、鈴木豪.「コンピュータ数学シリーズ 7 ゲーム計算メカニズム - 将棋・囲碁・オセロ・チェスのプログラムはどう動く -」, コロナ社, 2010.

受賞

1. Best Paper Award, 19th International Conference on Automated Planning and Scheduling (ICAPS'09), 2009.
2. 第23回人工知能学会全国大会優秀賞, 2009.
3. Best Paper Award, 13th Advances in Computer Games, 2011.