

詰将棋におけるプロセッサ稼働率を考慮したゲーム木並列探索

立松 靖朗^{†,☆} 山田 雅之[†]
世木 博久[†] 伊藤 英則[†]

我々は、詰将棋におけるゲーム木探索が並列に行えることに注目し、並列優先探索により詰将棋を解くプログラムを作成し、詰将棋問題における並列探索の有効性を示した³⁾。引き続き、本論文ではこの並列探索において、プロセッサを効率よく使用する方法を検討し、この方法を用いた場合の評価について述べる。その結果、プロセッサ稼働率を考慮しない従来の場合に比べ、解答能力、解答速度ともに向上することが確かめられた。

Improvement of Processor Efficiency in Parallel Search for *Tsume-Shogi* Game Tree

YASUO TATEMATSU,^{†,☆} MASASHI YAMADA,[†] HIROHISA SEKI[†]
and HIDENORI ITOH[†]

We have implemented a program for solving *Tsume-Shogi* problem based on a Parallel Best-First Search method, and have shown the effectiveness of the parallel search in *Tsume-Shogi* problem in 3). In this paper, we describe a method for improving the processor efficiency of the parallel search. Our program is improved by using this method. Its solvability and performance are evaluated.

1. はじめに

我々は、詰将棋におけるゲーム木探索^{1),2)}が並列に行えることに注目し、並列優先探索により詰将棋を解くプログラムを作成し、詰将棋問題における並列探索の有効性を示した³⁾。しかし、プロセッサの並列使用効率に課題を残した。そこで本論文では、プロセッサを効率良く使用することにより、解答能力と処理時間をさらに改善することを試みる。

従来のプログラム³⁾は、最初、ホストプロセッサが初手を求め、初手の数が n なら n 台のセルプロセッサに各々の初手以降の探索を依頼する方式をとる。この方式では初手の数がセルの総数より少ないときには、探索を依頼されないセルが生じる。これらのセルはしばらくの間待ち状態となり、これではセル全体の稼働率が低下してしまう。たとえば、内藤国雄問題集全 180 問⁴⁾では、初手数が 7~12 手の問題が多く、初

手数の平均は 10.33 である。また、20 手以下の問題が 9 割を占める。セル台数がこれ以上ある並列計算機では、初期において多くのセルが待ち状態となる。そこで本論文では、初手数がセル数より小さい場合でも各セルに仕事が行き渡るように、ホストが初手のみでなく初手以降のゲーム木も展開する場合の解答能力と処理時間を評価する。なお、並列計算機には、富士通 AP1000 を使用した。

2. プロセッサ稼働率を考慮した並列探索

ここでは、プロセッサ稼働率を考慮した並列探索方式について述べる。この方式では、初めにホストが探索を依頼するセルの数に下限値 n_s を設ける。初手の数が n_s に満たない場合、ホストは横型最良優先探索により新たな部分木を展開する（以降、ホストによる追加探索と呼ぶ）。 n_s 個以上の王手葉ノードをホストで生成し、その王手以降の探索をセルに依頼する（図 1 参照）。

ホストの処理：ホストの処理を図 2 に示す。

1. 最初、ホストは問題の初期局面が与えられると、すべての初手を求める。初手の数が n_s に満たな

[†] 名古屋工業大学

Nagoya Institute of Technology

[☆] 現在、日本電装クリエイト株式会社

Presently with Nippondenso Create Co.,Ltd.

- い場合は、横型最良優先探索によりゲーム木を展開する（追加探索）。ホストはこれを王手葉ノードの総数が n_s 以上になるまで行う。ただし途中解答が得られれば、その時点で処理を終了する。
2. 展開したゲーム木の王手葉ノード以降の探索を評価値の良いものから順に待ち状態セルに依頼する。
 3. セルによって生成されたゲームの部分木を受信し、自分自身が持っているゲーム木にこれを連結して新たなゲーム木を再構築する。ここで、受手ノードの一方で詰まないと判断された場合、または王手ノードの一方で詰むと判断された場合、これらのノードの兄弟ノード以下は無意味な探索部分

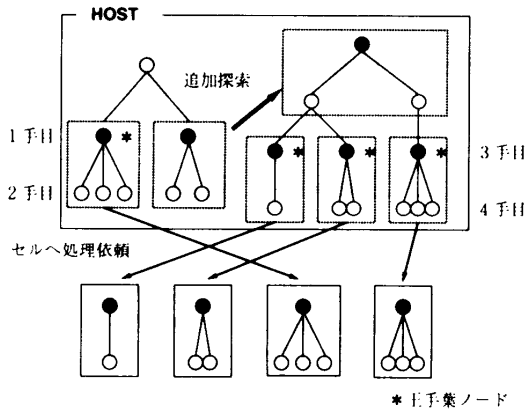


図1 ホストにおける追加探索
Fig. 1 A supplement search in Host.

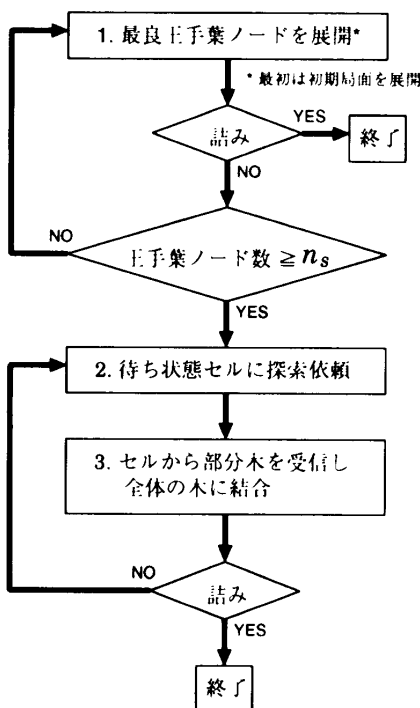


図2 ホストの処理
Fig. 2 Host process.

木であることから、これらの部分木の探索が割り当てられているセルが存在すれば、その処理を中断するメッセージをホストから該当セルに送る。ゲーム木を再構築した時点で解答が得られれば処理を終了する。そうでなければ、このゲーム木を MAX/MIN 法に基づき再評価し、2へ戻る。

セルの処理：セルでは横型最良優先探索を行う。セルは、(a) 受け取った王手以下のゲーム木が詰みであると判断できるまで、(b) 受け取った王手以下のゲーム木が不詰であると判断できるまで、(c) 生成した部分木の大きさがあらかじめ定められた量に達するまで、(d) ホストから処理中断のメッセージが到着するまでの、いずれかの条件を満たすまで処理を続ける。(a), (c) の場合は探索した部分木を、(b), (d) の場合は処理の終了メッセージのみをホストに送信し、新たな探索要求を待つ。

3. 評価

ここでは、プロセッサ稼働率を考慮することによる効果と作成したプログラムの解答能力について述べる。なお、セルは32台使用した。

3.1 プロセッサ稼働率を考慮することによる効果

プロセッサ稼働率を考慮した場合と考慮しない場合のセル稼働率を「続・詰むや詰まざるや」⁵⁾の第46番45手詰の問題を例にして比較する。図3に(a)従来（追加探索をしない）プログラムと(b)本（追加探索する）プログラムのセルの稼働状況を示す。横軸は時間であり、各々のグラフの縦軸は稼働セル台数である。従来プログラムと本プログラムのセル稼働率および処理時間は、それぞれ37%、27秒と、45%、20秒である。追加探索の効果によりセル稼働率が上がり、その結果、処理時間が短くなる。

表1は内藤国雄問題集と「続・詰むや詰まざるや」の問題における解答時間を示す。3, 5, 9, 15, 23手詰の問題では、ホストは正解に至る初手を選び、それ

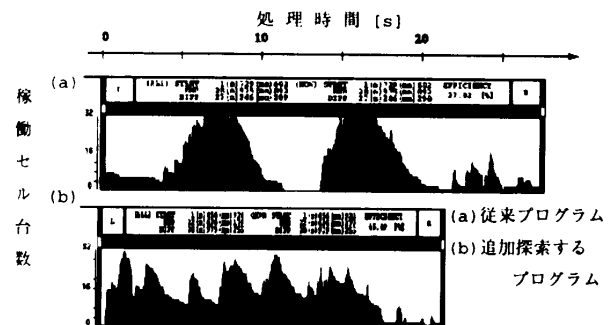


図3 セル稼働率状況
Fig. 3 The cell's busy trace.

表1 追加探索による解答時間の短縮
Table 1 Reduction in solving time by supplement search.

詰手数	解答時間	
	従来プログラム	本プログラム
3手	0.25	0.02
5手	0.27	0.04
7手	0.27	0.32
9手	0.62	0.36
15手	7.95	3.72
23手	6.08	5.02
27手	6.27	10.67
73手	18.56	18.60

以降を追加探索した。それ以外の問題では正解でない初手以降を追加探索した。正解手以降を追加探索した場合は全体の探索時間が短縮される。これは、正解手以降の探索に多くのセルが割り振られるためである。逆に、正解でない手以降を追加探索した場合、多くのセルに無駄な処理をさせるため、かえって解答時間が長くなる。

3.2 解答能力

ここでは、初期割当て用セル台数の下限値 n_s の最適値を実験により求めるとともに、他のプログラムとの解答能力の比較について述べる。なお、問題は「続・詰むや詰まざるや」全200問とした。また、使用メモリは64MB（仮想メモリ200MB）とし、1題30分の時間制限を与えて実験を行った。

3.2.1 初期割当てセル台数の下限値 n_s と解答能力

図4に n_s を変化させたときの解かれた問題の数を示す。総セル数は32とし、 n_s の値は12~28の範囲で実験を行った。解かれた問題の数は $n_s=20$ ($20/32=63\%$) のときに最大となった。 n_s の値がそれ以下のときには、セル稼働率が十分上がらないことが解答能力の向上を妨げたと考えられる。また、 n_s の値が20を越えるときに解答能力が低下する理由は、ホストの追加展開が過度に行われ、仕事待ち状態のセルが存在せずホストに蓄えられた仕事をセルに割り当てられなかったことによると考えられる。

3.2.2 他プログラムとの比較

図5に本プログラムにより解かれた問題の割合を詰手数別に示す。また、その結果を追加探索のない従来のプログラム、伊藤・野下⁶⁾らのItoおよびT2のプログラムの結果と比較する。追加探索の効果により、解かれた問題の割合が従来のプログラムに比べ全体に向上している。しかし、101手以上の問題では、ホストのメモリオーバーが原因で、本プログラムの方が解けない問題が多かった。その結果、解かれた問題の割合が従来プログラムより劣っている。本プログラムで

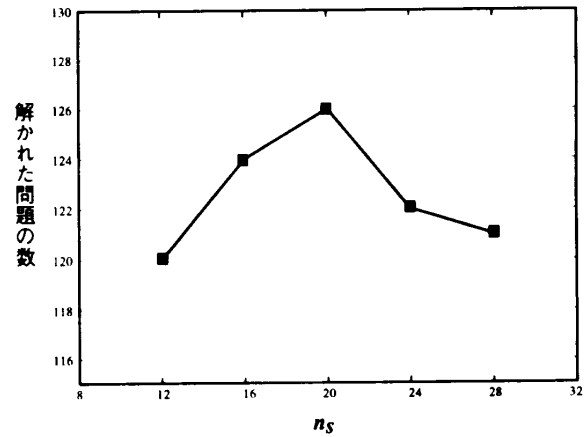


図4 初期割当てセル台数の下限値 n_s と解かれた問題の数
Fig. 4 Number of initial working cells n_s and number of solved problems.

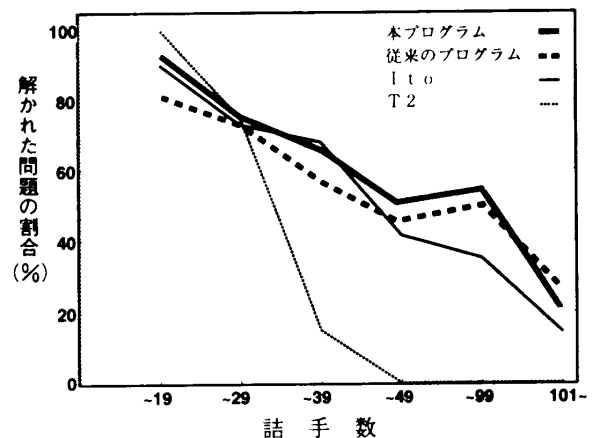


図5 「続・詰むや詰まざるや」詰手数別の解かれた問題の割合
Fig. 5 The rate of solved problems for each number of moves in Zoku Tsumuya Tsumazaruya.

は、セル稼働率を上げて並列度の高いゲーム木展開を実現している。この方式では、特に詰手数の大きい問題の場合、いらなくなった部分木を削除しメモリ領域を確保する処理を効率よく行う必要がある。しかし、この処理については従来のままであるためメモリオーバーとなる場合が多かったと考えられる。

また、並列探索の効果と、プロセッサ稼働率を考慮した効果により、単純な評価関数を用いているにもかかわらず、 T_2 と同等かそれ以上の解答能力が得られた。

4. おわりに

本論文では、詰将棋におけるゲーム木の並列優先探索においてプロセッサ稼働率を考慮したホストの追加探索による方式を提案し、その効果を示した。初期割当てセル数 n_s の最適値を求め、その値のもとで解答能力、処理時間を評価した。その結果、プロセッサ稼

働率を考慮しない場合と比べ解答能力、処理時間ともに向上した。しかしながら、評価値により追加探索するノードを決定し、それを優先して探索するため、追加探索したノードが正解手でない場合は無駄な処理が従来のプログラムより多くなる。その結果、解答時間が長くなったり、従来のプログラムで解けていた問題が解けなくなる場合がある。その中には、ホストのメモリアオーバーが原因のものも多い。これを解決するため、今後はセルを階層化し、一部のセルをメモリ媒体として利用する方法や、縦型探索による方法を検討する。さらに、詰将棋における探索に最も適しているとされる C^* アルゴリズム⁸⁾を並列化し、それとの比較についても検討する必要がある。

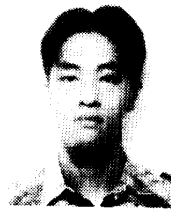
謝辞 論文をまとめるにあたり、ご協力いただいた現在日本 IBM 社の笠田洋和氏と和田誠一郎氏に感謝いたします。

参考文献

- 1) 松原 仁: 将棋とコンピュータ, 共立出版(1994).
- 2) 小谷善行, 吉川竹四郎, 柿木義一, 森田和郎: コンピュータ将棋, サイエンス社(1990).
- 3) 笠田洋和, 山田雅之, 松波功二, 世木博久, 伊藤英則: 詰将棋におけるゲーム木の並列探索とその評価, 情報処理学会論文誌, Vol.36, No.11, pp.2531-2539 (1995).
- 4) 内藤国雄: 九級から一級までの詰将棋, 成美堂出版(1992).
- 5) 門脇芳雄: 続・詰むや詰まざるや, 平凡社(1978).
- 6) 伊藤琢巳, 野下浩平: 詰将棋を速く解く2つのプログラムとその評価, 情報処理学会論文誌, Vol.35, No.8, pp.1531-1539 (1994).
- 7) 和田誠一郎, 立松靖朗, 山田雅之, 世木博久, 伊藤英則: 詰将棋プログラムにおけるゲーム木の並列探索手法の一考察, 第52回情報処理学会全国大会論文集(1996).
- 8) 脊尾昌宏: C^* アルゴリズムによる AND/OR 木の探索および詰将棋プログラムへの応用, 情報処理学会 AI 研究会, AI99-14 (1995).

(平成8年4月1日受付)

(平成8年6月6日採録)



立松 靖朗

1996年名古屋工業大学工学部知能情報システム学科卒業。現在、日本電装クリエイティブ勤務。



山田 雅之 (正会員)

1992年名古屋工業大学工学部電気情報工学科卒業。1994年同大学院工学研究科博士前期課程修了。同年同大学工学部知能情報システム学科助手。人工知能学会会員。



世木 博久 (正会員)

1979年東京大学工学部計数工学科卒業。1981年同大学院工学系研究科修士課程修了。同年4月より三菱電機(株)中央研究所に勤務。1985~1989年(財)新世代コンピュータ技術開発機構に出向。1992年4月より名古屋工業大学工学部知能情報システム学科助教授。工学博士。論理プログラミング、演繹データベース等に興味を持つ。電子情報通信学会, 人工知能学会, ACM, IEEE Computer Society 各会員。



伊藤 英則 (正会員)

1974年名古屋大学大学院工学研究科博士課程電気・電子専攻満了。工学博士号取得。同年日本電信電話公社入社。横須賀研究所勤務。1985年(財)新世代コンピュータ技術開発機構出向。1989年より名古屋工業大学教授。現在知能情報システム学科所属, 人工知能学会理事。これまでに、数理言語理論とオートマトン, 計算機ネットワーク通信 OS, 知識ベースシステムなどの研究と開発に従事。電子情報通信学会, 人工知能学会, ファジィ学会各会員。