

## 探索の重複領域を削減した階層的挟み撃ち探索による 実行時間最小マルチプロセッサスケジューリング問題の求解

### A Reduction of Overlapping Search Space for Hierarchical Pincers Attack Search for Minimum Execution-Time Multiprocessor Scheduling Problem

中村 あすか<sup>†</sup>  
Asuka Nakamura

富永 浩文<sup>†</sup>  
Hirobumi Tominaga

前川 仁孝<sup>†</sup>  
Yoshitaka Maekawa

#### 1. はじめに

実行時間最小マルチプロセッサスケジューリング問題の最適解求解手法としてPDF/IHS法(Parallelized Depth First/Implicit Heuristic Search)が提案されている[1]. PDF/IHS法は, CP/MISF法(Critical Path/Most Immediate Successors First)[2]のプライオリティリストを利用した分枝限定法を共有メモリ環境において階層的挟み撃ち探索で並列化する手法であり, 低コストの動的負荷分散や, 最適解を持つノードの探索木上の位置に関係なく高い高速化率を得ることができる. しかし, 本手法は, 並列化アルゴリズムに階層的挟み撃ち探索を用いるため, 左右から探索するプロセッサの探索が重複し, 無駄な探索が生じることがある. このため, PDF/IHS法は, 各ノードの分枝数が不定な探索木を生成するが, 各ノードの分枝数が少ない探索木において有効性が示された探索の重複領域削減手法[3]による無駄な探索の削減が有効であると考えられる. そこで, 本稿では, 各ノードの分枝数が不定な探索木を生成するPDF/IHS法に対する探索の重複領域削減手法の有効性を評価する.

#### 2. 実行時間最小マルチプロセッサスケジューリング問題

実行時間最小マルチプロセッサスケジューリング問題は, 処理時間および先行制約が任意のタスク $n$ 個からなるタスク集合 $T = \{T_1, T_2, \dots, T_n\}$ を処理能力の等しい $m$ 台のプロセッサからなるプロセッサ集合 $P = \{P_1, P_2, \dots, P_m\}$ で並列処理するスケジュールのうち, スケジュール長が最も短くなるスケジュールを求める問題である. ただし, 各プロセッサ間のデータ転送時間は無視できるほど小さく, 処理割込は起こらないとする. 本問題を, タスクをノード, 先行性約をエッジとしたタスクグラフで記述すると, DAG(無サイクル有効グラフ)で表される.

#### 3. PDF/IHS法

PDF/IHS法は, 逐次探索手法DF/IHS法(Depth First/Implicit Heuristic Search)を階層的挟み撃ち探索で並列化したスケジューリングアルゴリズムである. 以下の節では, DF/IHS法と階層的挟み撃ち探索について述べる.

##### 3.1 DF/IHS法

DF/IHS法は, CP/MISF法のヒューリスティックを用いた分枝操作を行う分枝限定法である. 本手法の分枝操作は, スケジュールが未設定となる時刻が最も早い

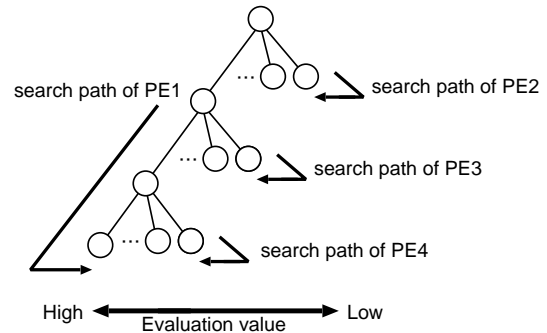


図1: 4PEによる階層的挟み撃ち探索の振舞

プロセッサに対して, スケジュールが未設定となる時刻に実行可能タスクの処理またはレディ状態を割り当てることで, 部分問題を生成する. このため, 各部分問題から生成される子問題数は, 実行可能タスクの個数によって変化するため, 本手法が生成する探索木は, 各ノードの子ノード数が一定にならない. また, 分枝操作で生成した部分問題は, CP/MISF法のプライオリティが高いタスクが割り当てられた部分問題ほど探索木左側になるように並び替える. これにより, 探索木左側から深さ優先探索を行うだけで, CP/MISF法のプライオリティが高いタスクを優先的に割り当てるスケジューリングパターンから順に探索することができる.

##### 3.2 階層的挟み撃ち探索

階層的挟み撃ち探索は, 複数のPE(Processor Element)が階層的に左右から挟み撃ち形で探索する共有メモリ環境向けの並列探索手法である. 本手法は, 1つのリーダーPEと複数のスレーブPEが並列に深さ優先探索を行う. 図1に, 階層的挟み撃ち探索における各PEの振舞を示す. 図中のPE1をリーダーPE, それ以外のPEをスレーブPEとする. リーダーPEは, 探索木左側から探索し, 自身が探索中のノードから根ノードまでを結ぶ経路上のノードを待ち状態のスレーブPEに割り当てる. スレーブPEは, 割り当てられたノードを根とする部分木を右側から探索する. このように, 下界の評価が良いノードを多くのPEで探索しながら下界の評価の悪いノードを少数のPEで探索するため, 下界の精度に影響されずに, 早い段階で精度の高いスケジューリングパターンを探索することができる.

本手法では, 同一階層を左右から2つのPEで探索するため, 同一ノードが2つPEに探索されることがある. これを, 探索の重複と呼ぶ. 同一ノードを複数回探索し

<sup>†</sup>千葉工業大学情報工学科, Department of Computer Science, Chiba Institute of Technology

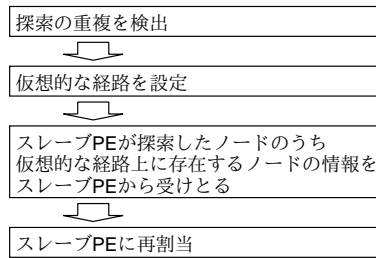


図 2: リーダ PE の探索の重複検出による再割当手順

でも、得られる結果は変わらないため、探索の重複が生じたノードに対する 2 回目以降の探索結果は無駄になる。無駄な探索を防ぐために、探索の重複を検出するための操作をスレーブ PE が行う。スレーブ PE は、探索の重複が検出すると待ち状態になり、リーダー PE から処理の再割当を受ける。

#### 4. 探索の重複領域削減手法を用いた PDF/IHS 法

PDF/IHS 法の探索の重複には、リーダー PE が先に探索するタイプ (A タイプ) と、スレーブ PE が先に探索するタイプ (B タイプ) の 2 パターンが存在する。階層的挟み撃ち探索では、スレーブ PE だけが探索の重複検出操作を行うため、A タイプの探索の重複が生じた場合は無駄な探索が生じないが、B タイプの探索の重複が生じると無駄な探索が生じる。筆者らは、分枝限定法の階層的挟み撃ち探索に対する探索の重複領域の削減による高速化手法を提案し、各ノードの分枝数が少ない探索木を生成する問題に対する有効性を示した [3]。PDF/IHS 法は、各ノードのスケジューリング時点で実行可能なタスクの数によってノードの分枝数が変化するが、探索の重複領域の削減手法を用いることで無駄な探索の削減を期待できる。そこで、本稿では、DF/IHS 法に対して探索の重複領域削減手法を用い、スレーブ PE と同様の探索の重複検出操作をリーダー PE も行うことで、B タイプの探索の重複によって生じた無駄な探索を削減する。探索の重複領域削減手法を用いた PDF/IHS 法のリーダー PE は、探索の重複を検出すると、探索が重複していたスレーブ PE に再割当を行う。図 2 に、探索の重複領域削減手法を用いた PDF/IHS 法のリーダー PE が探索の重複を検出した際の再割当手順を示す。ただし、図 2 中の仮想的な経路は、リーダー PE が探索中のノードを根とする部分木の最も左側を通る経路である。図 3 に、仮想的な経路の例を示す。リーダープロセッサが図中のスレーブ PE に再割当するノードは、根ノードからスレーブプロセッサが探索中のノードまでを結ぶ経路上に存在し、かつ、仮想的な経路上に存在するノード中で最も深さの深いノード B である。図 2 により、リーダー PE は、スレーブ PE の探索済ノード内の計算を再計算せずに探索することができる。また、探索の重複が生じたスレーブ PE は、自身が探索済みのノードを再探索することが無くなるため、階層的挟み撃ち探索の探索の重複領域を削減することができる。

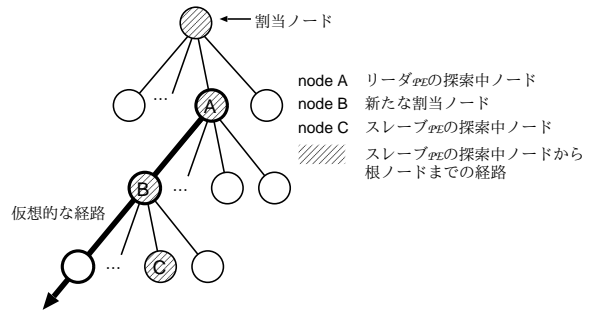


図 3: 仮想的な探索経路

表 1: 分枝数ごとのノード数

分枝数	ノード数
2	10
3	5
4	4
16	1

#### 5. 評価

PDF/IHS 法に対する探索の重複領域削減手法の有効性を示すために、DF/IHS 法が生成する探索木の各ノードの分枝数を評価する。表 1 に、タスク数 10 の実行時間最小マルチプロセッサスケジューリング問題求解において、DF/IHS 法が生成した探索木中のノードを分枝数ごとに示す。表 1 より、DF/IHS 法が生成する探索木には、探索の重複領域削減手法で探索ノードを多く削減することが期待できる分枝数の少ないノードが多く存在することが分かる。ここから、PDF/IHS 法に対しても、探索の重複領域削減手法が有効であると考えられる。

#### 6. おわりに

本稿では、探索の重複領域削減手法を用いた PDF/IHS 法の有効性を評価した。評価の結果、PDF/IHS 法に対しても、探索の重複領域削減手法を用いた PDF/IHS 法が有効であることが確認できた。

#### 参考文献

- [1] 笠原博徳, 伊藤敦, 田中久充, 伊藤敬介: 実行時間最小マルチプロセッサスケジューリング問題に対する並列最適化アルゴリズム, 電子情報通信学会論文誌 D, Vol. J74-D1, No. 11, pp. 755-764 (1991).
- [2] Hironori, K. and Seinosuke, N.: Practical Multi-processor Scheduling Algorithms for Efficient Parallel Processing, *IEEE TRANS. Comput.*, Vol. C-33, No. 11, pp. 1023-1029 (1984).
- [3] 中村あすか, 富永浩文, 前川仁孝: 探索の重複領域削減による階層的挟み撃ち探索の高速化, 先進的計算基盤システムシンポジウム論文集, pp. 348-355 (2011).