

3 J-5 実行時間最小・実用的並列最適化マルチプロセッサ・ スケジューリング・アルゴリズムの大規模問題への適用

飛田 高雄 笠原 博徳

早稲田大学理工学部電気電子情報工学科

1 はじめに

一般に強 NP 困難となることが知られてい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に対し、筆者等は、ヒューリスティックを用いた分枝限定法である応用した実用的な最適化アルゴリズム DF/IHS (Depth First / Implicit Heuristic Search) [1] を開発し、実問題に適用してその効果を確かめている [2]。さらに、DF/IHS の探索部分を並列化した PDF/IHS (Parallelized DF/IHS) を提案し、8 プロセッサマルチプロセッサシステム (ALLIANT 社 FX シリーズ) 上でインプリメントし、その効果を確認している [3]。本稿ではこの PDF/IHS を Kendall Square Research 社の KSR1 上でインプリメントし、タスク数数百から 1000 の大規模スケジューリング問題に適用した結果について述べる。

2 スケジューリング問題の定義

本稿で取り扱うマルチプロセッサスケジューリング問題は、能力の等しい m 個のプロセッサで、先行制約がありそれぞれの処理に要する時間が異なる n 個のタスクからなるタスク集合 T (タスクグラフと呼ばれる有限無サイクル有向グラフ(図 1)で記述される) をノンプリエンプティブに並列処理する際に、その実行時間(スケジュール長)を最小にするようなスケジュールを求める問題である。

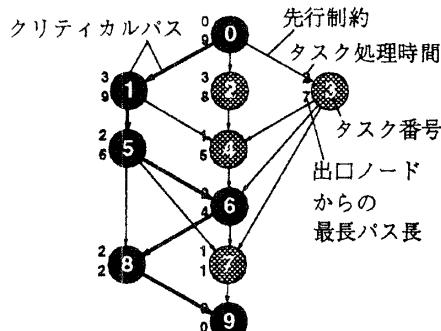


図 1: タスクグラフの例

3 実用的な並列最適化アルゴリズム

本節では上記スケジューリング問題に対する実用的な逐次最適化スケジューリングアルゴリズム DF/IHS [1] と、それを並列化した PDF/IHS [3] の概要を述べる。

Application of a Practical Parallel Optimization Algorithm for Minimum Execution-Time Multiprocessor Scheduling Problem to Large Scale Problems

Takao TOBITA, Hironori KASAHARA

Dept. of Electrical, Electronics and Computer Engineering,
School of Science and Engineering, Waseda University

3.1 DF/IHS

DF/IHS は、探索領域を低く抑えるために深さ優先探索を行ない、強 NP 困難性に対処するためにヒューリスティックを導入した、実用的な実行時間最小マルチプロセッサスケジューリングアルゴリズムである。DF/IHS は主に前処理部と探索部から構成される。前処理部では CP/MISF (Critical Path / Most Immediate Successors First) [1] と呼ばれるヒューリスティックアルゴリズムで使用されている割り当てプライオリティ、すなわち、タスクグラフ上で各タスクから出口タスクまでの最長パス長が長く、同一のパス長を持つタスク間ではより多くの直接後続タスクを持つタスクを優先するというものを使用して、探索の対象となる木の左側に良い解を集め。DF/IHS では、このプライオリティの高い順にタスク番号を付け替えるという作業を行なうことで、探索部で探索される木に上記のような特性を与える。この結果、分枝限定法を行なう探索部では、探索木の左側から右側へ単純な深さ優先探索を行なうだけで、プライオリティの高いタスクを優先してプロセッサに割り当てるスケジュールから順に探索でき、また、次分枝ノードの選択のためにヒューリスティック値を計算する必要がないので、効率の良いヒューリスティック探索が行なえる。さらに、探索の初期解が CP/MISF 解となるため、初期解がそのまま最適解となる場合が多いだけでなく、その他の場合でも、探索の初期に良い上限値(暫定解)が得られるため、Fernandez の下限値 [4] を用いて限定操作が効率的に行なえる。

3.2 PDF/IHS

PDF/IHS は、DF/IHS の探索木がヒューリスティック的に左側優先で重み付けられているという性質を用いて並列処理を行なう。以下に、 k 台のプロセッサエレメント PE を用いた場合の探索方法および探索領域割り当てについて述べる。

3.2.1 探索

PE_1 (リーダ) は逐次的な DF/IHS と全く同じように探索木の左から右へ探索を行う。他の PE_i ($i = 2, \dots, k$) (スレーブ) はリーダの探索経路上のノードを根ノードとして持つ部分探索木を右から左へと探索を行う(図 2)。すなわち、リーダとスレーブにより階層的な挟み撃ちの形で、各 PE は独立に深さ優先探索を行う。ただし、探索木の左端の初期解は精度の良い探索上限値となり得るため、冗長な探索を行わないように、 PE_1 が初期解を見つけ、その値が下限値(これ以上短いものはあり得ないというスケジュール長)に一致しないことを確認するまで、他の PE は探索を行わない。

PDF/IHS の探索では、ヒューリスティックが有効でなく、すなわち探索木の右側に最適解が存在し、DF/IHS で探索に長時間を要する問題に対しても、スレーブ PE がその解を見つけることができ、その場合には、探索時間を飛躍的に短縮し、スーパー リニアスピードアップを実現することが可能となる。また、PDF/IHS では、最適解に対する深さ優先探索の特徴として、並列化した際に減速異常が起こらない [5]。

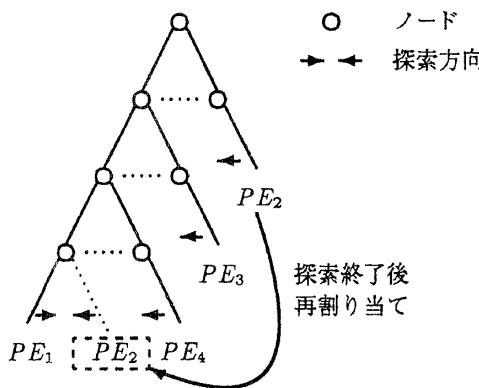


図 2: 階層挟み撃ち探索

3.2.2 探索領域の割り当て

PDF/IHS における探索領域の割り当ては、 PE_1 が PE_1 の探索経路上の深さの浅いノードを根として持つ部分木から順に、割り当て待ち状態にある PE に割り当てるという方法を用いる。このため、割り当て領域と割り当て深さが 1 対 1 に対応するので、次割り当て領域の指定が割り当て深さを指定するだけで簡単に実行できるので、スケジューラのオーバーヘッドを小さく抑えることができる。また、スレーブ PE が割り当てられた領域の探索を開始する際には、 PE_1 の探索経路から探索に必要なデータをほぼ独立に各スレーブ PE で再生できるため、領域割り当て後には PE 間のデータ転送を行なうことなく並列処理が可能である。

4 性能評価

本章では、PDF/IHS を分散共有キャッシュ型マルチプロセッサシステム Kendall Square Research 社の KSR1 上にインプリメントし、スケジューリングを行なった際の処理時間の計測結果について述べる。KSR1 では 32MB の Local Cache が階層的な ALLCACHE ENGINE により共有されており、ここでは 32 プロセッサまでの 1 クラスタを用いて計測を行なった。

4.1 評価方法

以下では用語の混乱を避けるため、スケジューリング問題の中で使われるプロセッサを単にプロセッサと表現し、PDF/IHS の並列処理のために使用する KSR1 のプロセッサを PE と表現する。

評価にあたっては、タスク間の先行制約が一般形状であり、タスク数 50~500、平均タスク処理時間は 20[u.t.] で標準偏差 4[u.t.] の正規分布、各タスクの平均先行タスク数 1~6 個(ランダム)となるように、乱数を用いてタスクグラフを多数作成し、それぞれ 2、3、4 台のプロセッサにスケジューリングを行なった際に、いずれかのプロセッサ数においては下限値が CP/MISF 解と一致せず探索を必要とするタスクグラフを 100 例選び、この合計 300 例のスケジューリング問題に対し、KSR1 の PE を 2、4、8 台用いての PDF/IHS を適用し、その計算時間について以下の計算式より加速率を計算した。

$$\text{加速率} = \frac{\text{DF/IHS} \text{ を用いた場合の探索時間}}{\text{PDF/IHS} \text{ を用いた場合の探索時間}}$$

ただし、問題が強 NP 困難であることから、探索時間は 600 秒を上限とし、DF/IHS あるいは少数の PE を用いての PDF/IHS では時間内に最適解が得られなかつたが、多数の PE を用いれば時間内に最適解が見つかったという場合には、探索を打ち切った時間で最適解が求まつたものとして、加速

率を計算した。なお、8PE を用いても時間内に最適解が求まらない問題は存在しなかつた。

また、より大規模な例として、タスク数 1000、平均先行タスク数 10 個、その他の条件は同じとしたタスクグラフを 5 例作り、プロセッサ数 4、5、6 台(合計 15 通り)に KSR1 の PE を 4、8、16 台用いてスケジューリングした。

4.2 評価結果

前述の 300 例の問題では、表 1 の通り、加速率が PE 数以上となる現象(スーパーリニアスピードアップ)がよく引き出されている。また、平均加速率も 2PE で 6.93 倍、4PE で 26.41 倍、8PE で 57.61 倍と高い値になっており、PDF/IHS が DF/IHS に比較しても非常に高速な探索を可能としていることがわかる。

またタスク数 1000 の大規模問題に対しては、表 2 のように DF/IHS では制限時間内に最適解を得られない問題でも、 PE を増やせば求解確率が上がる事が確認できる。

表 1: 平均探索時間と加速率(タスク数 500 以下)

PE 台数	1	2	4	8
平均探索時間(秒)	275.4	39.7	10.4	4.8
平均加速率(倍)	1.0	6.9	26.4	57.6
最大加速率(倍)	1.0	1473.5	2876.3	8968.6

表 2: 制限時間内最適解求解率(タスク数 1000)

PE 台数	1	4	8	16
最適解求解率(%)	26.7	80.0	93.3	100.0

5 まとめ

本稿では、初期解が CP/MISF 解と一致せず、探索を必要とするようなタスクグラフに対して並列最適化マルチプロセッサスケジューリングアルゴリズム PDF/IHS を適用した場合の求解時間を KSR1 上で評価した。その結果、DF/IHS では求解に長時間を要した問題に対して平均で 2PE で 6.9 倍以上のスピードアップが得られただけでなく、大規模問題に対しても有効であることが確認された。

今後は、データ転送を考慮したスケジューリング問題に対する実用的な最適化スケジューリングアルゴリズムを開発していく予定である。

謝辞

本研究の一部は文部省科学研究費補助金(一般研究(C) No.07680372)の補助で行なわれました。また本研究にあたり KSR1 を使用させて頂いたキャノンスーパーコンピューティングに感謝致します。

参考文献

- [1] H. Kasahara, S. Narita, "Practical Multiprocessor Scheduling Algorithms for Efficient Parallel Processing", IEEE Trans. on Computers, Vol.C-33, No.11, Nov.1984.
- [2] H. Kasahara, S. Narita, "Parallel Processing of Robot Arm Control Computation on a Multimicroprocessor System", IEEE Journal of Robotics and Automation, Vol.RA-1, No.2, Jun.1985.
- [3] 笠原, 伊藤(教), 田中, 伊藤(敬), "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", 信学論, Vol.J74-D-I, No.11, Nov.1991
- [4] E. B. Fernandez, B. Bussell, "Bounds on the Number of Processors and Time for Multiprocessor Optimal Schedules", IEEE Trans. Comp., Vol.C-22, No.8, Aug.1973.
- [5] Li G. J., Wah B. W., "Coping with anomalies in parallel branch-and-bound algorithms", IEEE Trans. Comput., Vol.C-35, No.6, Jun.1986.