

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 PDF/IHS の性能評価

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
〒169 東京都新宿区大久保 3-4-1 Tel: 03-5286-3371
E-mail: {tobita,kasahara}@oscar.elec.waseda.ac.jp

あらまし 本論文では、強 NP 完全な最適化問題であ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する実用的並列最適化アルゴリズム PDF/IHS (Parallelized Depth First / Implicit Heuristic Search) の性能評価結果について述べる。Sun Microsystems 社のマルチプロセッサシステム Ultra Enterprise 3000 上でタスク数 50 から 1100 の大規模問題に適用した結果、wall-clock time で 600 秒以内に最適解を得られた問題の割合は逐次型アルゴリズム DF/IHS (Depth First / Implicit Heuristic Search) で 69.3% だったのに対し、3PE を用いた PDF/IHS では 80.8%、6PE では 82.8% と向上した。また 6PE で最適解が得られた問題の平均求解時間は 1PE で 107.5 秒であったのに対し、6PE で 5.8 秒となっており、6PE を用いて 1PE の平均 18.5 倍の速度向上を得るなど、スーパーリニアスピードアップが有効に引き出せることが確認された。

Performance Evaluation of a Practical Parallel Optimization Multiprocessor Scheduling Algorithm PDF/IHS

TAKAO TOBITA HIRONORI KASAHARA

Dept. of Electrical, Electronics and Computer Engineering, Waseda University
3-4-1 Ohkubo Shinjuku-ku, Tokyo 169, Japan Tel: +81-3-5286-3371
E-mail: {tobita,kasahara}@oscar.elec.waseda.ac.jp

Abstract This paper describes performance evaluation of a practical parallel optimization algorithm PDF/IHS (Parallelized Depth First / Implicit Heuristic Search) to very large scale problems with more than one thousand tasks. PDF/IHS has been implemented on a shared main memory multi processor system Ultra Enterprise 3000 and applied to 50 - 1100 tasks problems. Within 600 second wall-clock time, it was confirmed that optimal schedules were found for 69.3% of problems on one processor, 80.8% on 3 processors and 82.8% on 6 processors. An average search time for optimal solution was reduced from 107.5 [s] by sequential DF/IHS to 5.8 [s] (18.5 times speedup) with 6 processors.

1. はじめに

マルチプロセッサシステム上で効率よく並列処理を行うには、タスク集合をどのようにプロセッサに割り当て、どのような順序で実行すべきか、という問題を解決しなければな

らない。この問題は実行時間最小マルチプロセッサスケジューリング問題^{1),2)}と呼ばれ、一般に強 NP 完全な極めて難しい最適化問題である¹⁾。

この問題に対し筆者等は、タスク数 300 程度の問題を数分で求解できる実用的な逐次型

最適化アルゴリズム DF/IHS (Depth First / Implicit Heuristic Search) を提案している²⁾。

しかし、このスケジューリング問題は強 NP 完全であるため、DF/IHS を用いても最適解の探索に長時間を要する個別問題が存在する。そこで、このような問題の求解所要時間を減少させる方法として筆者等は、DF/IHS を並列化した PDF/IHS (Parallelized DF/IHS) を提案している³⁾。本論文では、この PDF/IHS の性能を Sun Microsystems 社の Ultra Enterprise 3000 上で評価し、従来世界でも試みられなかったことがないタスク数千以上の超大規模問題に対しても数分以内に求解できることを確かめた。

2. 実用的最適化アルゴリズム

本論文で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 個のタスクの集合 $T = \{T_1, T_2, \dots, T_n\}$ を割り込みなしに並列処理する際に、そのスケジュール長を最小にするスケジュールを求める問題である。このタスク集合 T は、図 1 (a) のようなタスクグラフと呼ばれる DAG (有限無サイクル有向グラフ) で記述される。

2.1 DF/IHS

DF/IHS は、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探索順の決定のためにヒューリスティックを特殊な手法で利用した分枝限定法に基づく逐次型最適化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、前処理部では CP/MISF (Critical Path / Most Immediate Successors First)²⁾ 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を使用して、探索木の左側に良い解を集める。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ことで、簡単に探索木に上記のような特性を与えられる。この結果、分枝限定法による探索を行う探索部では、探索木の左側から右側へ単純な深さ優先探索を行う

だけで CP/MISF プライオリティの高いタスクを優先するスケジュールから探索でき、また、次分枝ノードの選択にヒューリスティック値を再計算する必要がなく効率のよいヒューリスティック探索が行える。さらに、探索の初期解が CP/MISF 解となるので初期解がそのまま最適解になる可能性が高いだけでなく、その他の場合でも探索の初期に現在ヒューリスティック的に得られる最良の上限値 (暫定解) が得られるので、Fernandez によって改良された Hu の下限値⁴⁾ 等の複数の下限値を用いて限定操作が効率的に行える。

図 1 (b) に、図 1 (a) のタスクグラフを DF/IHS により 2 プロセッサにスケジューリングする際の探索木を示す。図中の各ノードは、左側がプロセッサ 1、右側がプロセッサ 2 のタスク処理順序を表すガントチャートである。

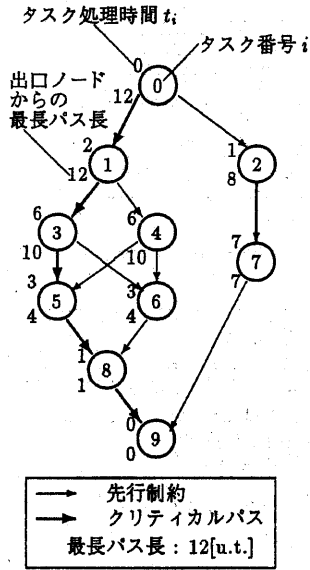
本論文で扱う問題では各タスクの処理時間が異なることを仮定しており、最適解を得るために、プロセッサをアイドルにする組合せも含めて探索する。

また、DF/IHS ではある時点でのレディタスクテーブルの中で何番目の位置にあるタスクをアイドルプロセッサに割り当てるかを示す SP (Selection Pointer) 値と呼ばれるポインタを持ち、次ノードの選択は辞書式順序で次に大きい値に SP 値を変更するだけで CP/MISF プライオリティの高いタスクから順に割り当てることができると共に、領域複雑度を低く抑えることができる。

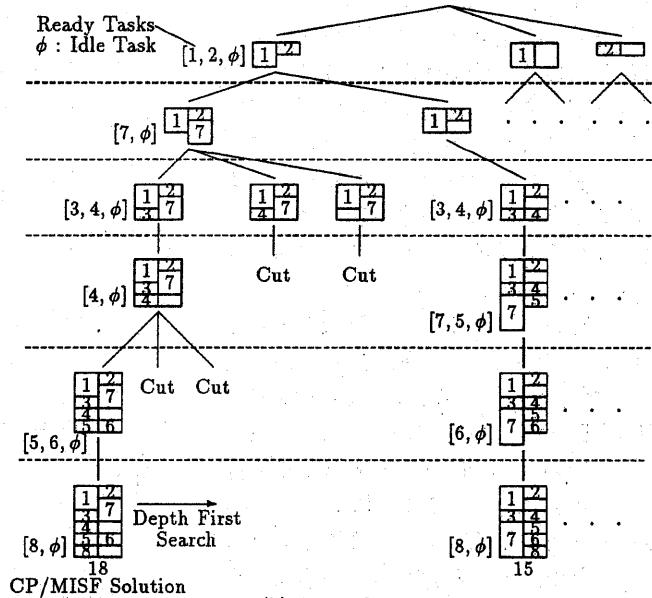
DF/IHS では、CP/MISF と同じ初期解が求まった後、下限値と比較してより良い解が存在する可能性があれば左から右へ探索木を深さ優先探索していく。

2.2 PDF/IHS

PDF/IHS は、DF/IHS をマルチプロセッサ上で並列化する際、並列探索にしばしば見られる減速異常 (複数プロセッサによる並列探索により逐次処理よりも長い時間を要してしまう現象)⁵⁾ を避け、できるだけスーパーニアスピードアップを引き出すと共に、部分探



(a) A task graph



(b) A search tree

図1 タスクグラフとそのDF/IHSにおける探索木

探索木をプロセッサへスケジューリングする際のオーバーヘッドを低く抑えつつ、プロセッサ間の負荷バランスを均等化することを目的に開発された並列探索アルゴリズムである。

PDF/IHSは、上記の目的のため、DF/IHS前処理によりヒューリスティックに良い解が集められた探索木の左側に多くのプロセッサが割り当てられ、スケジューリングオーバーヘッドを最小化しつつ負荷均等化をはかり、さらにヒューリスティックがうまく機能しない場合でも効果的な探索が行えるように階層的挟み撃ち探索方式を用いている。

この階層型挟み撃ち探索法では、 k 台のPEを用いる場合、 PE_1 (リーダー) が減速異常が生じないようにDF/IHSと同様に探索木の左から右に探索を行い、他の $PE_i (i = 2, \dots, k)$ (スレーブ) はリーダー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挟み撃ちの形式で探索する。すなわち、リーダーとスレーブは左右から階層的な挟み撃ちをし、スレーブは割り当てられた領域の探索を終了するまでリーダーと出会うまでは独立に探索

可能である。ただし、探索木の左端の初期解は現在最良のヒューリスティック解で精度のよい探索上限値 (暫定解) 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、その値が下限値に一致しないことを確認するまで、スレーブは探索を開始しない。

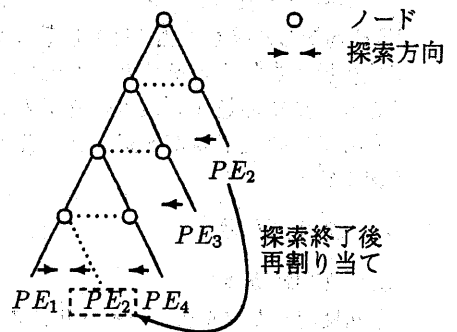


図2 各PEへの探索領域の割り当て

図2に、4台のPEを用いてPDF/IHSを実行する場合の各PEへの探索領域の割り当て

例を示す。各 PE への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探索木から順に、割り当て待ち状態にあるスレーブ PE に割り当てていく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行え、スケジューラのオーバーヘッドを小さく抑えられる。また、スレーブ PE が割り当てられた領域の探索を行う際には、次探索領域の割り当てを待つ間に PE_1 がブロードキャストした SP 値を使って探索に必要なデータを独立に各スレーブ PE が再生するため、PE 間のデータ転送量を低く抑えることができる。また、図 2 は、 PE_2 が最初の割り当て領域の探索を終了して待ち状態になった後、まだ未割り当ての領域のうち最も浅いノードを根に持つ部分領域に PE_2 の再割り当てを行った例を示している。

PDF/IHS における探索では、同一の部分探索木を PE_1 と PE_i ($i \neq 1$) が互いに反対方向に独自に深さ優先探索していくため、その探索木のどこかで 2 つの PE が出会うことになる。このような PE 間の出会いは、リーダの SP 値を各スレーブ PE にブロードキャストし、各スレーブ PE 内で PE_1 の SP 値と各自の SP 値の比較を行うことで独自に検出できる。出会いを検出したスレーブ PE はリーダ PE に通知した後割り当て待ち状態になり、まだ並列探索を行える部分木があれば再割り当てされ、全ての領域の探索が終了するか、どれかの PE の上限値が下限値と一致するまで探索を行う。

3. 性能評価

3.1 評価条件

今回使用したマルチプロセッサシステム Ultra Enterprise 3000 (以下 E3000 と略す) は、6 台の CPU と 384MB の共有メモリが 1 本のバスで接続されている主メモリ共有型マルチプロセッサシステムである。図 3 に、そのアーキテクチャ構成を示す。

以下では、スケジューリング問題の中で

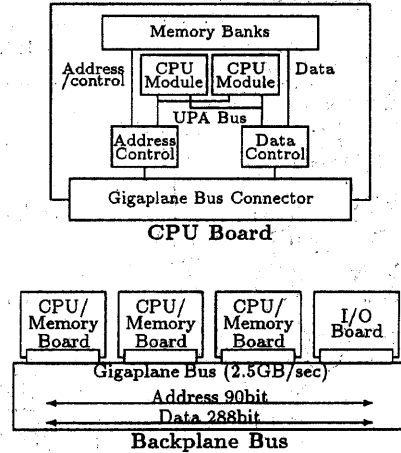


図 3 Ultra Enterprise 3000 のアーキテクチャ構成図

使われるプロセッサをプロセッサと表記し、PDF/IHS の並列処理のために使用される E3000 のプロセッサを PE と表記する。

本論文では、ランダムタスクグラフを作成し、E3000 上で DF/IHS および PDF/IHS を適用してスケジューリングした。各ランダムタスクグラフの生成条件は、タスク数 50~200 のもの (小規模) を 300 例、400~600 のもの (中規模) を 50 例、900~1100 のもの (大規模) を 50 例作成し、それぞれのタスク数において、平均先行タスク数を 2~10、平均タスク処理時間を 5~25、スケジューリング問題のプロセッサ台数を 2~10 の範囲でランダムに変化させた。取り扱うスケジューリング問題が強 NP 完全であり、PDF/IHS でも最適解を得るのに数日、数カ月かかる可能性もあり得るので、探索上限時間を 600 秒とした。

3.2 評価結果

上記の条件で作成したランダムタスクグラフ (400 例) に DF/IHS 及び PDF/IHS を適用した結果、600 秒の制限時間内に DF/IHS では 277 例 (全体の 69.3%) に最適解を得たが、3PE での PDF/IHS では 323 例 (80.8%)、6PE での PDF/IHS では 331 例 (82.8%) の問題に最適解が得られた。また、表 1 に 6PE で最適解が求まった問題についての加速率を

表1 6PEでの加速率 α

type	case	PE=1		PE=6	
		平均時間(秒)	平均時間(秒)	平均加速率(倍)	最大加速率(倍)
$\alpha \leq 1$	232	1.4624	1.6848	0.8680	0.9913
$1 < \alpha \leq 6$	27	126.2651	35.9826	3.5091	5.2111
$6 < \alpha$	72	442.2920	7.8471	56.3640	16048.6016
Total	331	107.5331	5.8230	18.4671	16048.6016

示す。ここで、

$$\text{加速率}\alpha = \frac{\text{DF/IHSの計算所要時間}}{\text{PDF/IHSの計算所要時間}}$$

である。3PEを用いたPDF/IHSにより最適解が得られた問題323例の1例当たりの計算所要時間の平均はDF/IHSでは95.3秒であるのに対し、3PEのPDF/IHSでは8.2秒となっており、DF/IHSと比較した加速率 α は3PEで11.6倍となっている。同様に表1のように6PEを用いたPDF/IHSにより最適解が得られた問題331例の1例当たりの計算所要時間の平均はDF/IHSでは107.5秒、6PEのPDF/IHSでは5.8秒となっており、加速率 α は6PEで18.5倍となっている。このように、PDF/IHSではスーパーリニアスピードアップが有効に引き出されている。当然、1PEでの処理時間を通常のDF/IHSではなく、PDF/IHSを1台のPEで実行した場合の値にした時には、スーパーリニアスピードアップは出現せず極めてリニアに近いスピードアップが得られる。ただしその場合には1PEでの求解時間が多くの場合DF/IHSより遅くなってしまうため、ここでは通常1PEで最高速のDF/IHSとの比較を行っている。

また、上記の平均所要時間は、3PEでは最適解が求まったが1PEでは最適解が得られなかった問題(46例)、表1における6PEでは最適解が求まったが1PEでは最適解が得られなかった問題(54例)については制限時間(600秒)で最適解が得られたものとして加速率を計算しているため、実際の速度向上はより大きな値となる。

また、DF/IHSおよびPDF/IHSは、本質的に分枝限定法であるため、許容近似精度 ϵ を設定し、厳密に暫定解 \hat{U} と最適解 t_{opt} との

相対誤差が ϵ 、 $0 \leq \epsilon \leq 1$ の解を見つけることが可能である。 ϵ を変化させていった場合の実行結果を表2に示す。この表は、まず $\epsilon = 0$ として600秒の探索を行い、制限時間内に求解できなかった問題に対して $\epsilon = 0.01$ として探索を行うといったように、以下順に時間内に求解できなかった問題に対してのみ ϵ をより大きな値にして探索を行った結果である。表2のように、 ϵ を大きくすることで探索時間を短縮できることがわかる。また、6PEでは600秒の上限時間内に全問題の90%に誤差1%以内、約98%に誤差95%以内、400例全ての問題に対して誤差10%以内の解が得られた。

4. 結 論

本論文では、実用的な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を従来では解かれたことのないタスク数1100の超大規模問題に適用した。その結果、Sun Microsystems社のUltra Enterprise 3000上で600秒の上限時間内に3PEを用いて80.8%、6PEを用いて82.8%の問題に対して最適解を得ることができると共に、全問題に対して誤差10%以内の高精度の近似解を求められることを確認した。また1例当たりの平均求解所要時間も、1PEでDF/IHSが107.5秒以上要したものが6PEで5.8秒となり、平均加速率は18.5倍とスーパーリニアスピードアップが有効に引き出されていることが確認された。この結果から、PDF/IHSは従来の逐次型スケジューリングアルゴリズムと比較して非常に短い時間で高精度の解を得られる有用性の高いアルゴリズムであることが確認された。今後の課題としては、プロセッサ間のデータ転送、処理とデータ転送のオー

表 2 PDF/IHS による解の誤差

近似誤差 ϵ		PE 1 台	PE 3 台	PE 6 台
$\epsilon = 0$	問題数	277 例	323 例	331 例
	割合	69.25%	80.75%	82.75%
	平均求解時間	11.4206 秒	8.1852 秒	5.8230 秒
$0 < \epsilon \leq 0.01$	問題数	60 例	38 例	33 例
	割合	15.00%	9.50%	8.25%
	平均求解時間	11.4364 秒	23.9612 秒	18.4363 秒
$0.01 < \epsilon \leq 0.05$	問題数	56 例	32 例	29 例
	割合	14.00%	8.00%	7.25%
	平均求解時間	14.0139 秒	4.7189 秒	2.5404 秒
$0.05 < \epsilon \leq 0.10$	問題数	7 例	7 例	7 例
	割合	1.75%	1.75%	1.75%
	平均求解時間	0.0037 秒	0.0065 秒	0.0075 秒
Total	問題数	400 例	400 例	400 例
	割合	100.00%	100.00%	100.00%

バーラッピングを考慮に入れたスケジューリング問題が挙げられる。

参考文献

- 1) Coffman, E. G.: *Computer and Job-shop Scheduling Theory*, John Willey & Sons (1976).
- 2) Kasahara, H. and Narita, S.: Practical Multiprocessor Scheduling Algorithms for Efficient Parallel Processing, *IEEE Trans. Comput.*, Vol. C-33, No. 11, pp. 1023-1029 (1984).
- 3) 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, *信学論*, Vol. J74-D-I, No. 11, pp. 755-764 (1991).
- 4) Fernandez, E. B. and Bussel, B.: Bounds on the Number of Processors and Time for Multiprocessor Optimal Schedules, *IEEE Trans. Comput.*, Vol. 22, No. 8, pp. 745-751 (1973).
- 5) Li, G. J. and Wah, B. W.: Coping with anomalies in parallel branch-and-bound algorithms, *IEEE Trans. Comput.*, Vol. C-35, No. 6, pp. 568-573 (1986).