

疎結合リング型マルチプロセッサにおける Branch&Boundの並列処理

4L-4

青柳 雄大 緑川 博子 飯塚 肇
成蹊大学 工学部

1. はじめに

最適解を求めるためBranch & Bound法 (B & B) を行う必要性がしばしばあるが、一般に非常に時間がかかる。そこで筆者らは、マルチプロセッサによる並列B & Bアルゴリズムを提案し、性能評価を行った。

2. アルゴリズム

ここで用いたアルゴリズムは、基本的には図1 aのように2進木の一方の部分木をある1台のプロセッサで担当し、他方を他のプロセッサに担当させるという処理を、各レベルで行うものである。各プロセッサは、割り当てられた部分木を、深さ優先で解いていく。自分の割り当てられた部分木の探索を終了すると、まだプロセッサに割り当てられていない部分木を図中の数字の順に担当し、残された部分木がなくなるまで処理する。

またパラレベルPと呼ぶパラメタを導入し、プロセッサの集中の度合を制御できるようにした。例えば、図1 a (P = 1) では、初めから木全体を複数のプロセッサで分散的に調べ始める。図1 b (P = 3) では、まず右の木から集中的に複数のプロセッサを投入し、右の木の割り当てが終了すれば、次第に左の木に探索を広げていく。対象とする問題で探索順が右あるいは左から行う方が有利であるという性質がわかっている場合、このPによって一方向から集中的に並列処理していくことができる。また使用するプロセッサ台数に応じて、適当なPを設定することもできる。

このアルゴリズムは、最低1台は逐次処理で行う探索経路をたどるので、逐次処理より遅くならない。また使用プロセッサ台数は自由に設定可能で拡張性がある。

3. 性能評価

本実験では、トランスピュータT800 9台 (内1台はホスト) からなる疎結合リング型のマルチプロセッサで処理を行った。扱った問題は100石のナップサック問題である。

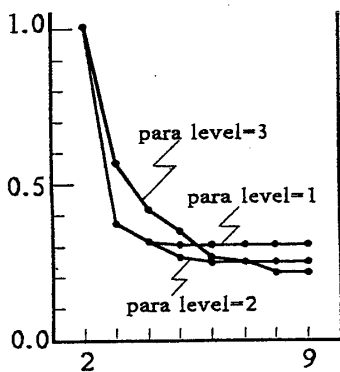
図2 a、図2 bにプロセッサ台数の違いによる逐次処理に対する処理時間の変化を示す。図2 aは平均的な例で、図2 bは加速異常現象がみられた例である。それぞれ逐次処理の約5.4倍、5.6倍の性能が得られた。

また100題の問題をランダムに作成し、9台のプロセッサを用いて処理時間の平均をとると、逐次的の場合の約3倍の性能であった。プロセッサの台数効果が得られない理由の1つは、逐次探索と並列探索では探索空間が異なるからである。即ち、並列探索では、逐次探索では探索しない部分木もあらかじめ探索してしまう場合がある。反対にプロセッサ相互で下界値を更新し合うと、boundingが早く、逐次探索に比べ探索空間を縮小できる場合もある。これが加速異常現象の例で、使用プロセッサ台数以上の性能が得られる理由である。

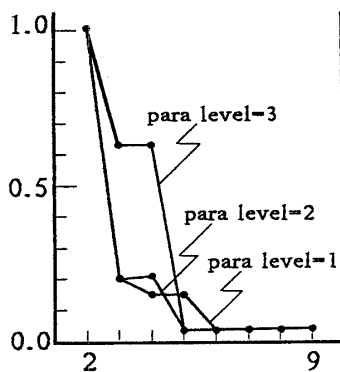
また探索木の構造によって、プロセッサ台数分の部分木生成が追いつかなかったり、葉に近づくにつれて割り当てが頻繁になり、再割り当てのための通信時間のオーバーヘッドが増える為、プロセッサ稼働率が低下する。

4. おわりに

今回の実験では、アルゴリズムのチェックとおおまかな性能の試算の為、ナップサック問題を行ったが、通信オーバーヘッドが相対的に小さくなる一般の整数計画問題について今後性能評価をおこなう予定である。



プロセッサ台数
図2 a 逐次探索に対する
並列探索相対処理時間
(加速異常現象のない場合の例)



プロセッサ台数
図2 b 逐次探索に対する
並列探索相対処理時間
(加速異常現象のある場合の例)

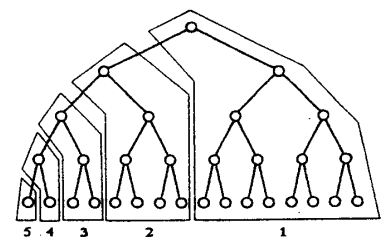


図1 a 部分木と割り当ての順序
(P = 1 の場合)

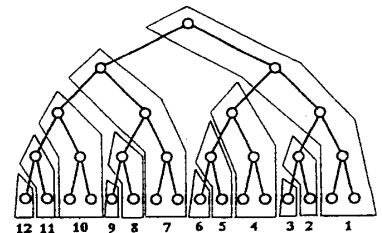


図1 b 部分木と割り当ての順序
(P = 3 の場合)