2T－10

マシーンスケジューリング問題における並列アルゴリズムと実現

小林俊彦　中森眞理雄
東京農工大学 工学部 情報工学大講座

１．はじめに

組合せ的な制約条件のもとで、目的関数を最大や最小にする組合せ最適化問題の一つとしてマシーンスケジューリング問題がある。この問題を解くための有効な解法である分枝限定法によってこの問題を解くと、部分問題がそれぞれ独立した問題として扱われるため、並列化に適していると考えられる。そこで、マシーンスケジューリング問題における並列分枝限定法のアルゴリズムを設計し、分散メモリ型並列計算機である nCUBE2 上で３２台のプロセッサを用いて実現した。

２．マシーンスケジューリング問題

マシーンスケジューリング問題は、m 個のマシーンで q 個の品物を処理する問題である。各品物は、あらかじめ定められた順序（処理の流れ）に従って、マシーンによるオペレーションを経なければならない。各マシーンは、同時に二つのオペレーションを実行することはできない。各オペレーションの実行時間は与えられている。オペレーションの数は n とする。

マシーンスケジューリング問題は、選択グラフによって記述することができる。選択グラフには、処理の流れ全体の始点と終点があり、あるオペレーション i と j が同一の処理の流れで隣接していれば、C弧と呼ばれる枝で接続される。オペレーション i と j が同一のマシーンでの処理で、異なる処理の流れに属しているならば、D弧と呼ばれる枝で双方向に接続される。ここですべてのD弧の組のうち一方向の枝を選択し、閉路ができないような有向グラフを作ることにより、マシーンがオペレーションを処理する順序を決定する。マシーンスケジューリング問題は、このような方法でできる多数の有向グラフの中で、最小のクリティカルパスを有するものを求める問題である（図１）。

A parallel algorithm and its implementation for the machine scheduling problem
TOSHIHIKO KOBAYASHI , MARIO NAKAMORI
Tokyo University of Agriculture and Technology

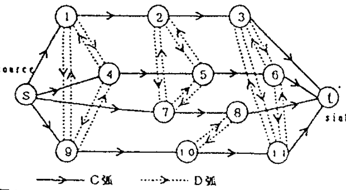

図１．マシーンスケジューリング問題の選択グラフによる記述

３．並列分枝限定法

３．１　アルゴリズムの並列化

並列化は、一台のプロセッサが他のプロセッサの状態を把握するマスタスレーブモデルをもとに考えた。

(1) 準備処理

- マスタプロセッサがファイルからＣ弧の接続情報とオペレーションの実行時間、Ｄ弧の初期状態を読み込み、前の二つの情報を各プロセッサに送信。
- マスタプロセッサが初期問題を解き暫定解を得たら、それをスレーブプロセッサに送信。

(2) マスタプロセッサ（すべてのスレーブプロセッサの状態を把握）の処理

- 自分が部分問題を持ちフリーのプロセッサが存在する場合は、部分問題を分与。
- 部分問題がなくなったら、そのことをスレーブプロセッサに通知。
- 暫定解が更新されたら、すべてのスレーブプロセッサにその値を送信。

(3) スレーブプロセッサの処理

- マスタプロセッサに部分問題がない場合、部分問題を持つものは、マスタプロセッサに部分問題を送信。
- 現在解より良い暫定解を得たら、それをマスタプロセッサに送信。

以上の処理を並列に実行させることにより、nCUBE2 上で並列分枝限定アルゴリズム実現した。

３．２　部分問題の管理

それぞれのプロセッサにおける部分問題の管理は、待機行列を用いている。問題を解きいくつかの部分問題が発生したら、その中の一つの部分問題を次に解く部分問題とし、残りは待機行列の最後部に入れる。こ

のとき、部分問題が発生しない場合や、限定操作が起きた場合は、次に解く部分問題を待機行列の最後部から取り出す。また、マスタープロセッサがスレーブプロセッサに部分問題を渡す場合と、その逆の場合は、待機行列の最前部から問題を取り出して渡す。この方式によって、それぞれのプロセッサが探索を行うときは縦型探索、別のプロセッサに問題を渡す場合は横型探索を行うようにした。

4．結果

ncube2 上で３２台のプロセッサを用いて、逐次処理との比較を行った。与えた問題は、マシーン数 m 、品物の個数 q を固定し、オペレーションの実行時間と各マシーンが実行するオペレーションの数を変えた５０題である。逐次処理の計算時間をT_1、プロセッサがP台の時の計算時間をT_Pとすると、速度向上率は$S_P = T_1 / T_P$で表される。逐次処理での部分問題の数と、P＝３２の場合に得られた速度向上率の平均との関係を表１に示し、問題の散布図を図２に示す。

表１．部分問題の数と速度向上率の関係

問題数	5000 未満	5000～ 9999	10000～19999	20000～29999	30000～39999	40000～49999
速度向上率	14.25	17.08	19.38	19.56	21.3	22.17

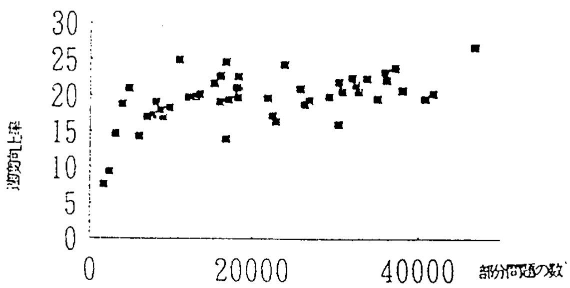

図2．問題の散布図

表１より、部分問題の数が多くなるほど速度向上率が良くなり、３２台のプロセッサを使用することによって、２０倍程度の速さを得られることがわかる。部分問題の数が少ない問題は、非並列アルゴリズムの時の部分問題の数よりも多い部分問題を解くことが多いために、台数効果が得られない。

次に比較的速度向上率が悪かった問題において、各プロセッサにおける計算時間と通信時間の割合を図３に示す。

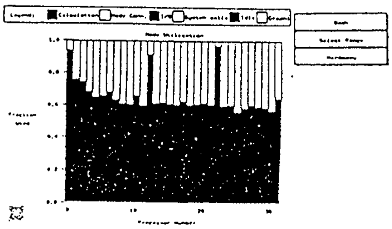
図3．通信と計算の割合

図３のように、数台のプロセッサに部分問題が集中してしまうと、他のプロセッサの通信回数が増え、速度向上率はあまり上がらないことがわかる。

5．おわりに

今回の結果より、プロセッサに部分問題がうまく均等されないことが多いことがわかった。それぞれのプロセッサが持つ部分問題の数には、ばらつきが大きく、問題の最後の方（探索木の終端に近い方）では、全体のプロセッサ数に対して少ない数のプロセッサに部分問題が集中して残ることが多い。しかも、問題の最後の方で発生する部分問題は、ほとんどが限定操作の対象となってしまう。そのようなスレーブプロセッサが存在して、マスタプロセッサが部分問題を持っていないと、このアルゴリズムでは、それらのプロセッサ間で部分問題のやり取りが行われることになる。しかもその部分問題がほとんどが限定操作の場合、子問題もできなく、少ない計算時間で終了してしまうため、通信のオーバーヘッドが大きくなってしまう。

これを改善するには、プロセッサ間でまとめて部分問題を転送できればよいが、部分問題となるこのD弧のデータ構造の場合、５０組のD弧が存在すると、一つの部分問題に対して 1.5k バイトもの、大きさになってしまう。そのため、各プロセッサの待機行列はリスト構造にする必要があり、問題をまとめて送ることができない。そこで、部分問題を表すD弧のデータ構造の改善を行う必要がある。

参考文献

[1] 高橋義造，“並列処理機構”，丸善株式会社，1989.
[2] 岡本秀輔，渡辺一衛，飯塚肇，“フローショップスケジューリングにおける総所用時間最小化問題に対する並列アルゴリズムとnCUBE2上の実現”，電子情報通信学会技術研究報告(SS92-13)，1992.