

グリッド環境下における処理順制約を考慮した タスクスケジューリングの最適化

森田 亘, 平松 綾子, 能勢 和夫

大阪産業大学工学研究科

1. はじめに

近年、注目されているグリッド・コンピューティングでは、大規模な計算ジョブを構成する複数のタスクをどのコンピュータ (計算リソース) に割り当てるのかというスケジューリングが重要な問題となる。従来、グリッド環境下での様々なスケジューリング手法が考案されている[1][2]。しかしながら、これらの手法は、タスク間に連携性のない独立したタスクであることが前提とされている。そのため、タスク間に連携性のある場合には、連携性のあるタスクを大きな1つのタスクとして扱い、分散した計算リソースを有効活用することができない。

本稿では、タスク間に連携性のあるジョブに関するスケジューリング問題を対象とする。すなわち、リソース間に能力差等がある場合、および、タスク間の処理順に先行制約のある場合のタスクスケジューリング問題を制約条件付き組み合わせ最適化問題としてとらえ、アントコロニー最適化法 (Ant Colony Optimization : ACO 法) による解法を提案する。

2. 想定するグリッド環境

以下のようなグリッド環境を考える。

- ・スケジューリングは、マスタと呼ばれる管理マシンが行う
- ・マスタは、スケジューリングだけでなく、各計算リソースの管理も行う
- ・各リソースは、異種環境 (処理能力、メモリ容量が異なる) とする
- ・いくつかのタスク間には、処理順に関する先行制約があるものとする

3. 問題の記述

計算リソースを $m_1, m_2, \dots, m_M$ 、ジョブを構成するタスクを $r_1, r_2, \dots, r_N$ とする。リソース m_j の処理速度を v_j 、タスク r_i の処理量を q_i とする。タスク r_i がリソース m_j で処理されるとき処理時間は $t_{i,j} = q_i / v_j$ である。リソース m_j の容量を c_j 、タスク r_i の処理に必要なリソースの容量を s_i とする。もし $s_i > c_j$ なら、タスク r_i はリソース m_j に割り当てることができない。タスク r_i の処理完了時刻を ρ_i とすると、ジョブの処理完了時刻 F は、すべてのタスクの処理が完了した時刻

$$F = \max_{i \in M} \{\rho_i\} \quad (1)$$

となる。このとき、ここで論じるスケジューリング問題は、「ジョブの処理完了時刻 F が最早となるように、すべてのタスクについて処理リソースとそのリソースでの処理開始時刻を決めること」となる。

4. 最適化手法

4.1 最適化手法の概要

本稿では、タスクスケジューリングの最適化手法として、組み合わせ最適化手法の一つである ACO 法の適用を検討する。

アリがフェロモンに基づいて餌を探索する行動を模した多点探索法である。解候補の空間をノード空間とし、複数のアリエージェントが、ノード上に蓄積されたフェロモン量をたよりに最適解の探索を行う。

タスクスケジューリング問題に ACO 法を適用するときの基本的な考え方は次のとおりである。まず、解候補を表現するノード空間として、タスクの計算リソースへの割り当て順を決めるノード空間 (順序ノード空間) と割り当てリソースを決めるノード空間 (割り当てノード空間) を準備する。ノード選択の際に、順序ノード空間ではタスク間の先行制約を考慮し、割り当てノード空間ではタスクの容量制約を考慮するものとする。そして、ガントチャート上に時間的先行制約を考慮してタスクを前詰め配置し、各タスクの処理開始時刻を得る。

Optimization for Task Scheduling with Processing Order Constraints in Computational Grid
Wataru Morita, Graduate School of engineering, Osaka Sangyo University
Ayako Hiramatsu, Graduate School of engineering, Osaka Sangyo University
Kazuo Nose, School of engineering, Osaka Sangyo University

4.2 ノード空間

本手法で用いる解候補のノード空間を図 1 に示す。タスクのリソースへの割り当て順序を決めるノード空間は、図 1 の (a) に示すようなノード空間で、ノード (i, j) は、 j 番目に割り当てられるタスクが r_i であることを意味する。したがって、順序ノード空間は、一般的には、 $N \times N$ の正方形の空間となる。なお、一度選択されたタスク番号は、それ以降の処理において、選択候補から除外する。また、アリが選択できるのは、先行制約を満たすタスク番号のみである。これにより、アリがノード区間を左から右へ移動することにより、タスクの割り当て順序が決まる。タスクの割り当てリソースを決めるノード空間は、図 1 の (b) に示すようなノード空間で、ノード (i, j) は、タスク r_i をリソース m_j に割り当てることを意味する。したがって、割り当てノード空間は、一般的には、 $M \times N$ の長方形の空間となる。なお、アリが選択できるのは、容量制約を満たすリソース番号のみである。これにより、アリがノード空間を左から右へ移動することにより、各タスクの割り当てリソースがタスク番号順に決まる。

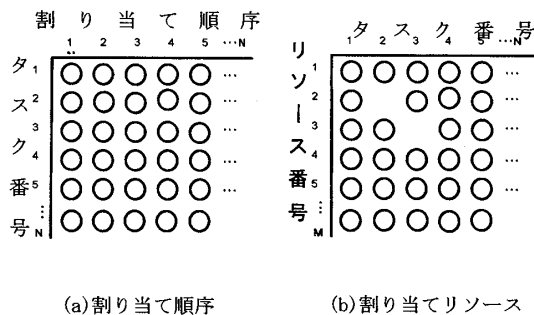


図 1. ノード空間

5. 実験結果

以下の条件で数値実験を行った。制約率とは、容量制約によって割り当てることができないタスクとリソースの組み合わせの割合である。

- ・ リソース 20 台
- ・ タスク 200、2000 個
- ・ アリ数 20 匹
- ・ 世代数 3000 世代
- ・ 制約 0%~40%

行った数値実験の結果を図 2、図 3 に示す。

制約率が大きいほど、全タスクの処理時間は増加する。また、問題が複雑になるため解の探索に要する世代数も増加すると考えられる。

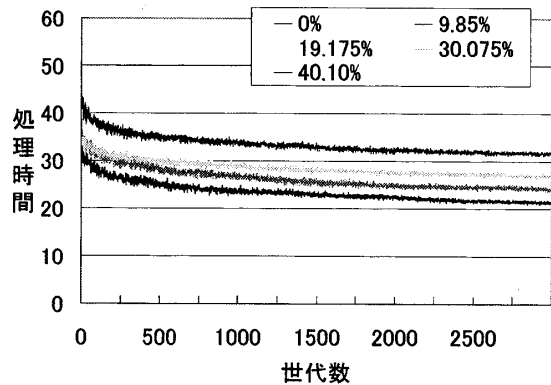


図 2. 制約率の比較(タスク数 200)

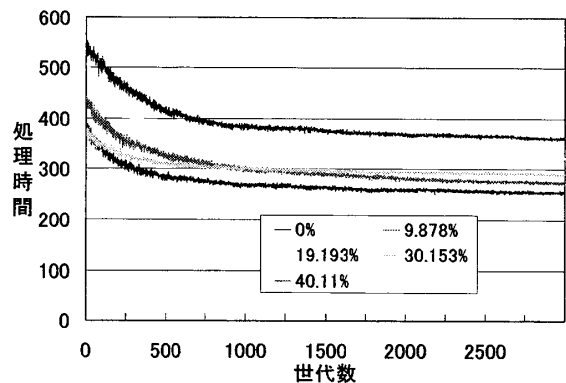


図 3. 制約率の比較(タスク数 2000)

6. まとめ

グリッド・コンピューティングにおけるジョブの分割タスクに関する制約条件付きスケジューリング問題について、ACO 法による解法を提案した。本手法では、順序ノード空間により、先行制約を満たすタスク順を決め、割り当てノード空間により、容量制約を満たす計算リソースへのタスクの割り当てを決めることで、タスク連携性を考慮したスケジューリングを実現する。メモリの容量制約を考慮したスケジューリングが可能であることを確認した。今後、連携性を踏まえた問題に対する数値実験を行う。

参考文献

- [1] Jing Liu, Li Chen, Yuqing Dun, Lingmin Liu, and Ganggang Dong: "The Research of Ant Colony and Genetic Algorithm in Grid Task Scheduling," Proc. of 2008 Int. Conf. on MultiMedia and Information Technology, pp. 47-49 (2008)
- [2] Yixiong Chen: "Load Balancing in Non-dedicated Grids Using Ant Colony Optimization," Proc. of Fourth Int. Conf. on Semantics, Knowledge and Grid, pp. 279-285 (2008)