これより、グリッドコンピューティングにおける、制約条件付きスケジューリング問題への、ACO法の適用の研究発表を行います。

研究者は、13H023恩田です。

次

グリッドコンピューティングにおいて重要になってくるスケジューリング問題について

先行制約、容量制約の制約条件を満たしつつ、最適化を行うことを目的とします。

全探索では非常に時間がかかるため、組み合わせ最適手法のACO法を用いて解を導き出すことにしました。

今回特にACO法の改良を行います。

次

スケジューリング問題について

まず、複数のジョブがあり、その中に多数のタスクが存在します。

そのタスクにはそれぞれ処理量が定められています。

また、マシンにもそれぞれに処理能力を定められています。

他に先行制約と容量制約があり、

これらの条件を考慮し、タスクをマシンに割り当てていくことを制約条件つきスケジューリング問題といいます。

また、結果が早く終われば終わるほどよい結果であると言えます。

次

先行制約と容量制約について、

先行制約について

1層のタスクがすべて終わると2層に、2層のタスクがすべて終わると3層、

このように前の階層のタスクが終わらないと次の階層のタスクに進めない、

という階層構造を先行制約といいます

次に容量制約です。

容量制約は、(先程あった)、マシンの処理能力とタスクの処理量を比較し、

図のような、タスクの処理量よりもマシンの処理能力が上回っていないと、

そのタスクを実行できないことを言います

次

本研究ではスケジューリングの最適解を求めるために、ＡＣＯ法を用います。

ＡＣＯ法はアリがフェロモンを介して餌を探す動きを模して考えられた集団探索アルゴリズムです。

本研究では解の候補をノード空間で表現し

処理順ノード空間では、ジョブ内のどのタスクを処理するかを、先行制約を満たした状態で決定します。

配置順ノード空間では、処理順により決まった各ジョブのタスクの処理順に基づき、すべてのジョブのすべてのタスクを配置する順番を決める。

割当ノード空間は、マシンがタスクを処理できるのか、つまり容量制約を考慮しながらマシンにタスクを割り当てます。

その解候補をガントチャートに表すことによって評価値として処理(完了)時間がわかります

（その処理時間に応じた）その評価値からフェロモンの蒸発と散布を行います。

さらにノードを蓄積フェロモン量をもとに確率的に選択して解候補を作ります。

アニメーション

この三つを繰り返し行い、その数を世代数とします。

今回の研究ではフェロモンの蒸発の改良を行います。

次

処理順ノード空間では、先行制約があるため、選択されることのないノードが存在します。また、配置順ノード空間でも各ジョブから、選択の対象外となるノードが存在します。

従来の方法では、制約条件によって選択されることのないノードでも蒸発を行ってきたのですが、今回は、選択されることのないノードでは、蒸発率を下げるように変更しました。

今回の数値実験は、この蒸発率を動的に変更することにより、どのような変化があるかを見ることを検証します。

次

蒸発率の適用箇所の変更

次

これはジョブの条件です

ジョブの数を5

それぞれのタスクの処理量は50~100

次

マシンの処理速度，容量はそれぞれを60~100の条件で10刻みに，乱数を用いてランダムに設定を行った

次

ＡＣＯの条件は

アリ数を50、世代数を10000まで、蒸発率は0.05と0.01の二つとし

それぞれ100回試行しました。

次

まず蒸発率0.05の結果です。

こちらは100回試行したときの、それぞれの試行ごとの最良評価値の出現頻度です。

また右上にあるのはその最良評価値の、最小値、最大値、平均値になります。

青色が処理ノード・配置ノードに蒸発率の変更をしているもので

オレンジが何も変更を加えていないもの

グレーが処理のみに、緑が配置のみに変更を加えているものです

縦軸と横軸の説明

値が小さければ良い結果となります

比較的、bとsが良い値のほうに出現回数が集まっています。

アニメーション

また平均値の部分をみてもこの二つが良い結果を出しているのがわかります。

次

次に蒸発率が0.01の結果です。

0.05ときよりも、bとsが良い結果なのがわかります。アニメーション

さらに平均値でも良い結果なのがわかります。

upver、haichiの結果が他のsyori,bever比べ悪い結果となった。

この問題設定では、蒸発の適用箇所の変更は最良解の改善にはつながらなかった。

配置の考察

次

容量制約がない場合

次

先ほどの数値実験から容量制約がどのような影響を与えているのか、検証しました。

マシンの条件を先ほどの条件から、容量をすべて100に設定しました。

これでマシンの処理能力に差はあるものの、すべてのタスクがこなせるようになります。

次

まず、処理完了時間が大きく減少していることがわかります

出現回数では先ほどの容量制約ありの結果と比べると、uとhの結果が他の2つよりよくなっています。

平均値では一番良い結果がuということになりました。

次

こちらでも、uとhの結果が、最良評価値アニメーション、平均値アニメーション、ともに良い結果となりました。

2つの環境において容量制約なしの方が、処理完了時間が大きく減少した。

upver、haichiの結果が容量制約があるものと比べ特に良くなった。

配置の考察

次

まとめ