

GA による ScaLapack のコスト見積もり関数を用いたスケジューリングの実装 斉藤 宏樹

1 今月の課題

GrADS プロジェクトが開発した ScaLapack のコスト見積もり関数を用いて、グリッド上でのスケジューリングを行った論文がある¹⁾。今月は GA を用いてこの論文と同様の実験を行い、論文で用いられていた Ad-Hoc greedy 法と SA によるスケジューリングとの性能比較を行った。

2 遺伝的アルゴリズムによるスケジューリング手法

ScaLAPACK の実行時間を見積もるために必要となる計算資源のスケジュールを GA の個体として表現して、最適な個体を選び出す。最適化されるスケジュール情報は、使用する計算機の組み合わせと、タスクを割り当てる計算機の順番である。以下に各個体のコーディング方法と、コーディングされた個体の評価方法について説明する。

2.1 個体のコーディング

グリッドで使用できる計算機 1 台を 1 設計変数とし、1 設計変数を 1 ビットの遺伝子で表現する。そのビットが 0 であればその計算機を使用せず、1 であれば使用するものとする。これより、使用する計算資源の組み合わせを個体の遺伝子の値によって決定することができる。選んだ計算資源の使用順序は、次節に記すように評価時に決定する。

2.2 個体の評価

個体の評価方法は以下の通りである。本手法では 1 個体につき 2 回評価を行うことになる。

1. 個体の遺伝子から得られたスケジュールをもとに、ScaLAPACK の解析プログラムから実行時間の見積もりコストを計算し、個体の評価値 f_1 を求める。
2. 1 で評価したスケジュールの計算機の使用順序を、ランダムに変更する。そして、再び ScaLAPACK の解析プログラムから実行時間の見積もりコストを計算し、個体の評価値 f_2 とする。
3. f_1 と f_2 を比較して良好な評価値をその個体の評価値とする。

3 数値実験

実装した GA によるスケジューリングの性能を検討するため、論文¹⁾で示された 2 種類のクラスタ、合計 14 ノードからなるグリッド上で、ScaLAPACK の実行時間の見積もりが最小となるスケジュールの探索を行った。

3.1 パラメータ

実験に用いた GA のパラメータを Table 1 に示す。

Table 1 GA parameters

parameter	value
Population Size	100
Chromosome Length	12 (=L)
Selection Method	Tournament
Tournament Size	4
Crossover Method	Two-Point Crossover
Crossover Rate	0.6
Mutation Rate	0.0833 (1/L)
Number of Elites	1
Number of Generations	500

3.2 Ad-Hoc greedy, SA との性能比較

各問題サイズにおいて、Fig. 1 の結果が得られた。Fig. 1 の結果から、GA の性能は各問題サイズにおいて Ad-Hoc greedy よりも良く、SA とほぼ同等の性能を示したことが確認できる。

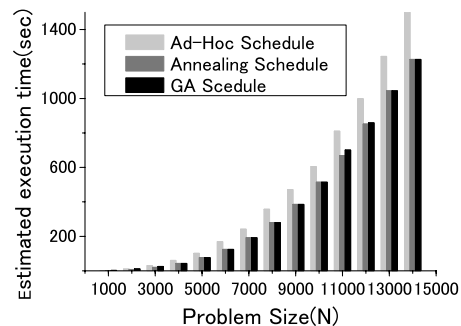


Fig. 1 複数計算資源における GA と Ad-Hoc greedy, および SA の見積もり実行時間の比較

参考文献

- 1) Asim YarKhan, Jack J. Dongarra. Experiments with scheduling using simulated annealing in grid environment. *Workshop on Grid Computing (submitted)*, June 7, 2002