

Grid 環境を想定した大規模ジョブ分配問題の実装および検討  
 斉藤 宏樹

1 今月の課題

今月の課題として、Grid 環境を想定した大規模ジョブ分配問題の実装および検討が挙げられる。

2 課題の進捗状況

2.1 Grid 環境を想定した大規模ジョブ分配問題の実装

Grid 環境を想定した大規模ジョブ分配問題とは、Grid のようなヘテロな環境において、大規模なジョブをどう最適に処理するか、その最適なジョブ分配を求める問題である。この問題を SGA で解く場合には、解を遺伝子型としてどのようにコード化するか、交叉や突然変異のオペレータとして何を採用するかといった問題がある。今月は、それらの問題を考慮した実装を検討した。

2.1.1 遺伝子型へのコード化

分配ジョブ 1 つ 1 つを設計変数とし、設計変数の遺伝子型は整数型 1 ビットで表現する。その遺伝子型の値はそのジョブを処理する CPU 番号とする。Fig. 1 に、その方法を示す。



Fig. 1 コーディング方法

Fig. 1 は、大きさの異なる 10 個のジョブを 5 台の CPU に分配する例である。CPU 番号 1 の処理時間を求める場合、個体の遺伝子型の値を全て検索し、“1”が割り当てられているジョブ番号を探す。そして、そのジョブ番号をもとに、リストを参照して得られる実際のジョブサイズから、CPU 番号 1 の処理能力で割るという操作をする (CPU の処理能力についても、CPU 番号をもとにリストを参照して得ることにする)。

2.1.2 交叉・突然変異のオペレータ

交叉法は遺伝子長に応じて変更する。短い場合は一点交叉を、長い場合は一様交叉を適用する。突然変異は遺伝子座の値を他の対立遺伝子の値に置き換える方法をとることにする。

2.2 数値実験

オーバーヘッドを含まない簡単なジョブ分配問題を SGA で解いた。

2.2.1 実験内容

式 (1) に示すジョブの処理時間  $T$  を最小にする最適なジョブ分配を求める。

$$T = \max\left\{\frac{x_1}{p_1}, \dots, \frac{x_{50}}{p_{50}}\right\} \quad (1)$$

$x_i$  は分配ジョブの大きさであり、 $p_i$  はプロセッサの処理能力である。プロセッサの処理能力は 10 ~ 500 まで 10 づつ大きくなる 50 種類に分かれており、それぞれ 1 個ずつ計 50 個利用できる。大規模ジョブの量を 12750 とする。用いた SGA のパラメータを Table 1 に示す。

Table 1 SGA のパラメータ

個体数	100	突然変異率	0.00078
遺伝子長	1275	エリート保存	1
交叉	一様	終了世代	2000
交叉率	1.0	試行回数	20

結果は Fig. 2, Fig. 3 のようになった。この問題設定では、あらかじめ CPU の処理能力と同等のジョブを分配すれば、最適値が 1 になるように設定しているの、最適値が得れていることがわかる。

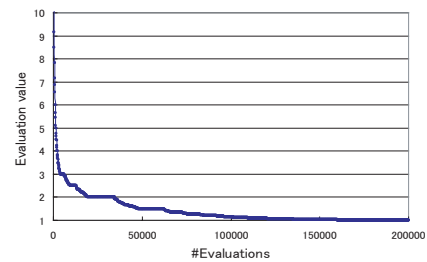


Fig. 2 評価関数値の推移

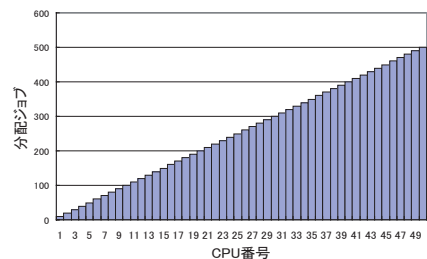


Fig. 3 最適解

3 翌月への課題

翌月への課題としては、大規模ジョブ問題の設定を複雑にし、今回実装した SGA で解くことが挙げられる。