

# 遺伝的アルゴリズムのためのグローバルコンピューティング機構について

齊藤大晃 新井浩志

千葉工業大学工学研究科

## 1. はじめに

遺伝的アルゴリズム (Genetic Algorithms; 以下 GA) は様々な分野に適用されており, その処理速度を改善するための手法として並列化の研究が行われている. 例えば, ネットワークで接続されたコンピュータクラスタに DGENESIS<sup>[1]</sup>を導入することによって手軽に並列 GA 処理を実現することが可能となる. しかし, 大規模なコンピュータクラスタを構築するためにはかなりの設備投資が必要である.

これに対して近年では, インターネット上の計算機資源を活用して並列処理を実現するグローバルコンピューティングに関する研究が行われている. GA に対するグローバルコンピューティングプロジェクト<sup>[2]</sup>も提案されているが, 巡回セールスマン問題専用であり, 一般的な最適化問題には利用できない.

本報告では, GA のためのグローバルコンピューティング機構を提案する. 本機構を実現することにより, インターネット上に接続された任意のマシンから並列 GA 処理を利用することが可能となり, 各種最適化問題の解を効率的に求めることができる.

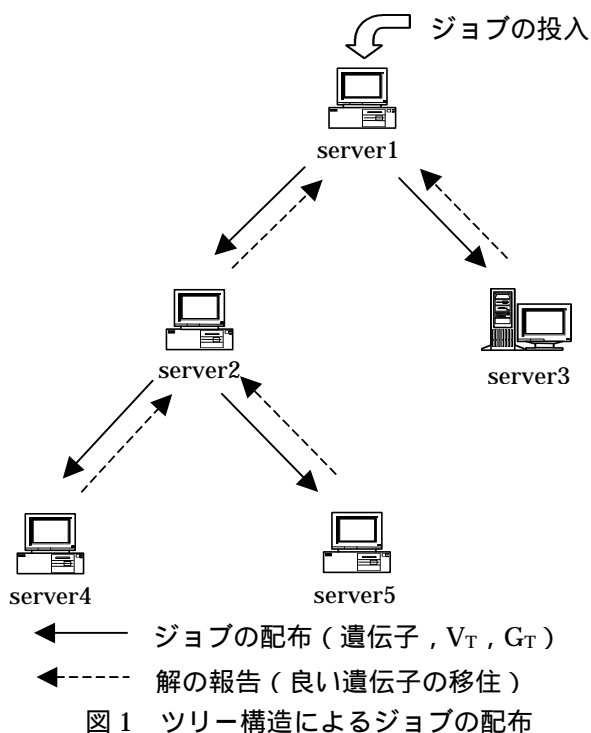
## 2. グローバルコンピューティング機構

### 2.1 ジョブの投入と配布

従来の多くのグローバルコンピューティングでは 1 台のホストマシンがジョブの配布と管理を行っており, マシンの負荷が集中してしまうという問題がある. このため, ホストマシンにはかなりの処理能力が要求される. また, グローバルコンピューティングに参加する個々のマシンは計算機資源を提供するのみであり, 自らがジョブの投入を行うことは認められていない.

本提案手法では, 論理的に隣接するマシンの一覧を各サーバマシンに登録して仮想的な網目構造を構築する. そして, ユーザはどのサーバからでもジョブの投入ができるようにする. ジョブの要求を受けたサーバは, 自身が GA 処理を行うと同時に隣接するマシンに接続して同一のジョブを配布する. この

ようにして, 投入されたジョブはツリー構造で各サーバマシンに配布されていく. この様子を図 1 に示す. このとき, 1 台のマシンが同一のジョブを重複して処理してしまうことを避けなければならない. そこで, 最初にユーザからジョブの要求を受けたサーバはユニークなジョブコードを生成し, 各サーバはこのジョブコードによりジョブを識別する.



### 2.2 移住の制御

ジョブ配布のツリー構造の上層側を親, 親によってジョブの配布を受けた下層側を子とする. 鳥モデルによる並列 GA<sup>[3]</sup>では, 移住間隔を定め, 一定の世代ごとに遺伝子を移住させている. これに対して本提案では移住間隔を固定せず, 子から親へ中間報告するタイミングで移住を行うものとする. 子は GA 処理によって得られた解の中間報告として, 良い遺伝子を親に移住させる. 良い解はなるべく早く親に報告することが望ましいが, 頻繁に接続するとネットワークの負荷が増大してしまう. そこで, このバ

ランスを取るために、各サーバ毎に目標評価値  $V_T$  と、親への報告期限世代数  $G_T$  を定義する。

子は親から指示された  $V_T$  と  $G_T$  を上限として GA 処理を開始し、 $V_T$  より良い評価値が得られるか、または  $G_T$  世代を経過する毎に、上位  $N$  個の遺伝子を親に移住させる。また、このとき同時に新しい  $V_T$ 、 $G_T$  の指示を受けるものとする。自分と自分の子に対しては次式 (1)、(2) に従ってより細かい単位で  $V_T'$ 、 $G_T'$  を設定して、移住処理を行う。

$$V_T' = V_{MAX}^n + (V_{MAX}^n - V_{MAX}^{n-1}) \times \frac{G_T}{g} \times \dots (1)$$

$$G_T' = G_T \times \dots (2)$$

ここで、前回の最良評価値を  $V_{MAX}^{n-1}$ 、今回の最良評価値を  $V_{MAX}^n$ 、実際に処理した世代数を  $g$  とする。

また、 $\alpha$  は  $V_T$  の増分を決める変数 ( $0 < \alpha < 1$ )、 $\beta$  ( $0 < \beta < 1$ ) は  $G_T$  を分割する変数である。

親から与えられた  $V_T$ 、 $G_T$  をもとに、 $V_T'$ 、 $G_T'$  を計算しながら GA 処理を進める場合の例を図 2 に示す。

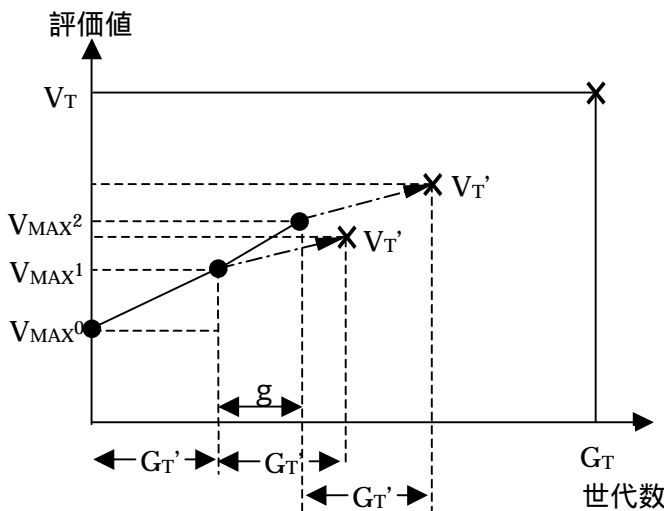


図 2 目標評価値の設定

### 3 評価と考察

10Mbps のイーサネットに接続された 6 台の PC (FreeBSD) 上に本提案のグローバルコンピューティングサーバを実装し、基本的な動作の確認を行った。GA を適用する問題としてはナップザック問題を用い、遺伝子長 70 ビット、遺伝子数 80 個で試行した。各サーバがジョブを分配する子の数の上限を 2 台、親への報告期限世代数  $G_T$  の初期値を 1600、

$V_T'$ 、 $G_T'$  を求めるのに使用した変数  $\alpha$ 、 $\beta$  をそれぞれ 0.5、0.25 とした。図 3 にサーバマシンの台数を 1 台、3 台、6 台と変化させた場合の実行結果を示す。1 台だけ使用した場合に比べ、台数を増やしたほうがより良い評価値を早く得られることがわかる。これは、本グローバルコンピューティング機構によって並列 GA 処理が行われ、移住によって進化が促進し、早く良い評価値を得ることができたためであると考えられる。

今後は数十～数百台規模で本手法の有効性を検証する予定である。

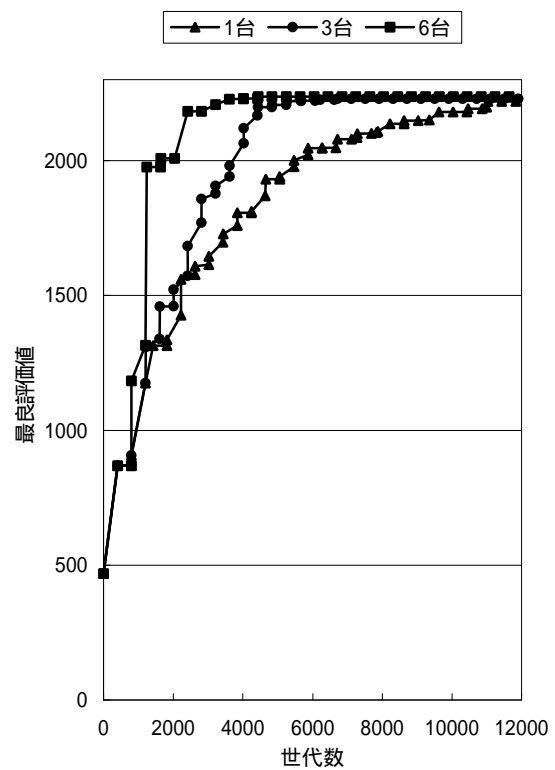


図 3 最良評価値の推移

### 参考文献

- [1] DGENESIS, <ftp://www.aic.nrl.navy.mil/pub/galist/src/dgenesis-1.0.tar.Z>
- [2] GeneticTSP, <http://www.ad-infinitum.net>
- [3] 三木光範, 畠中一幸, “並列分散 GA による計算時間の短縮と解の高品質化”, 日本機械学会, 第 3 回最適化シンポジウム講演論文集, 1998.