

# Grid 計算環境における遺伝的アルゴリズムのモデルの検討

## Genetic Algorithms on the Computational Grids

○学 谷村 勇輔（同志社大院） 正 廣安 知之（同志社大工）  
正 三木 光範（同志社大工）

Yusuke TANIMURA, Graduate School of Engineering, Doshisha University  
Tomoyuki HIROYASU, Mitsunori Miki, Department of Engineering, Doshisha University  
{tanisuke@mikilab, tomo@is, mmiki@mail}.doshisha.ac.jp

In this study, a model of Genetic Algorithms (GAs) on the computational grids is discussed through the numerical experiments. The proposed system is a master worker model. In this system, a master site reserves the information of the best searching point of a worker site in a queue when the master checks the calculation progress of the workers. At the same time, the master site sends the other information of the searching point that is derived in the worker site to the other worker site. By this operation, the searching information is transferred to every worker site. The proposed system can solve the problems on the computational grids; the sudden halt of the site and network. These characteristics of the proposed system are confirmed and discussed through the numerical experiments.

*Key word:* Grid Computing, Parallel Computing, Genetic Algorithms, Optimization

### 1 はじめに

近年、科学技術計算に代表される大規模計算を行なうためのポータルとして Grid 計算環境が注目されている。Grid は、広域に配置された複数の計算資源を結びつけて分散・並列計算を行なうことのできるシステムをユーザに提供する。現在、Grid を構築・運用するための基盤ソフトウェアが精力的に研究開発されており、テスト運用も開始されている。しかしながら、そのような Grid 環境が何に利用でき、どう利用するのかという応用研究は、NetSolve<sup>2)</sup>の研究などがあるが、ほとんど行われていない。

最適化計算は構造物の設計などの際に利用される技術であるが、解析を繰り返し行なうために概して計算コストが高くなる。これを解決する 1 つの方法が並列化であり、さらに Grid 環境を利用することができれば、大規模な最適化問題を解くための解決策となる。本研究では最適化手法の 1 つである遺伝的アルゴリズム (Genetic Algorithms, 以下 GA) に着目し、Grid 環境を利用できる GA のモデルを検討し、システムの構築を行なう。GA は多点探索で高い並列性を有している<sup>3)</sup>。同時に各探索点は比較的独立性の高い探索を行なうので、探索点の一部が失われても影響は小さい。これらは、Grid 環境を利用して最適化計算を行うのに適した特徴であるといえる。

本論文ではハイパークラスタ型の Grid 環境を想定し、マスタ・ワーカ型の GA モデルについて検討を行なった。モデルの検討は、疑似的に Grid 環境を構築し、ワーカとなっている計算機を停止させたり、

計算機間のネットワークを途切れさせたりするシナリオをもとに行なった。そして、検討モデルを実現するシステムを構築し、簡単な評価を行なった。

### 2 2 個体分散遺伝的アルゴリズム

GA は目的関数が連続関数の時のみならず、離散関数の時にも用いることができるため、幅広い応用が期待されている最適化手法である。GA では、各探索点を個体と呼び、個体の集合は母集団と呼ぶ。GA は母集団内の各個体に対して交叉や突然変異と呼ばれる遺伝的操作を適用し、新しい個体を生成する。この時点において、古い個体を親個体、新しい個体を子個体と呼ぶ。そして親個体、子個体の中から、現在の環境への適合度の高いものが次の世代に生き残る。GA では、これら一連の操作が行われる周期を世代と呼んでいる。こうして世代を重ねていくことにより優れた個体だけが生き残り、結果として最適解が得られるのである。

GA は優れた手法であるが、各個体の評価計算を多数行なうために解が得られるまでに膨大な時間が必要となる。そこで GA の並列モデルの研究がなされている<sup>1)</sup>。その 1 つに島モデルがある。本研究では、島モデルを拡張した 2 個体分散遺伝的アルゴリズム (Dual Individual Distributed Genetic Algorithms: Dual DGA) を利用する<sup>4)</sup>。Dual DGA は島内の個対数を 2 と設定することで、実装時のアルゴリズムを簡易化している。また Dual DGA は、できるだけ全体の多様性が維持できるように、交叉や突然変異を

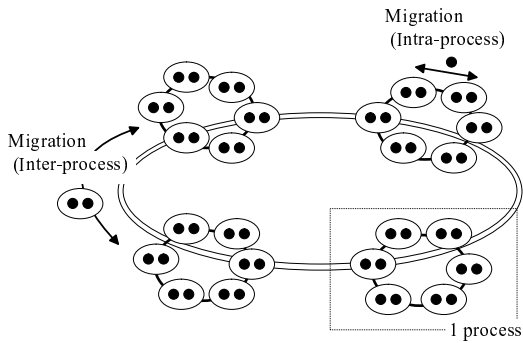


Fig. 1 Migration model of Dual DGA

独自の方法で行なう。その並列モデルは Fig. 1 に示すように 2 段階の階層型移住を採用し、並列計算機を利用する際にプロセッサ間の通信負荷を減らすことができるように工夫されている<sup>5)</sup>。

Dual DGA の利点は解探索の能力が高いだけでなく、並列モデルを実装しやすいという特徴をもつ。以下に Dual DGA の特徴をまとめる。

**[Dual DGA]**

- 従来の分散 GA に比べて解探索能力が高い。
- 分散 GA の仕組みを受け継いでおり、特に分散 GA を階層化したモデルが構築しやすい。

**[Parallel Dual DGA]**

- 通信量が少なく、高速に計算を行える。
- 1 つの島を粒度の単位とすることで、並列化の粒度を柔軟に変更できる。
  - － 実行する並列計算機の（通信／計算）の割合に合わせた粒度を用いることができる。
  - － ロードバランスがとりやすい。

**3 Grid GA システム**

**3.1 Grid 計算環境の特徴**

Grid 計算環境は従来の計算環境と異なる特徴をもつ。特にアプリケーション開発者は次のことを考慮して、計算資源を有効に利用するべきであると考えられる。(1) 各計算拠点がもつ計算能力、および各計算拠点を結ぶネットワーク性能は様々である。(2) また、それらの処理能力は時間とともに変化する。(3) 予測できない障害により、計算拠点が利用できなくなったり、ネットワークが切断される可能性がある。

**3.2 検討モデル**

本研究では 2 節の理由から、Dual GA を基本とした GA の Grid モデルについて検討を行なう。本研究では計算拠点として分散メモリ型の並列計算機を想定し、それぞれをサイトと呼ぶ。この時、ある 1 つのサイトは他サイトとコミュニケーションを行うための処理（管理プロセス）を実行する。これをマ

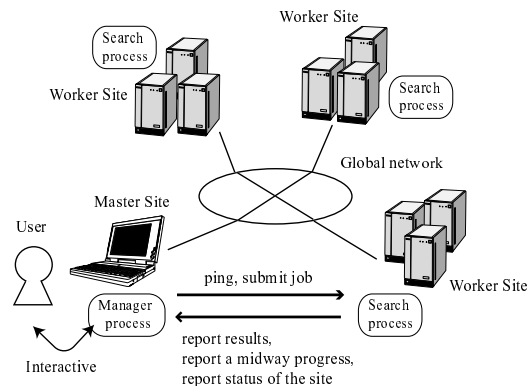


Fig. 2 Concept of the master-worker based GA on the computational grids

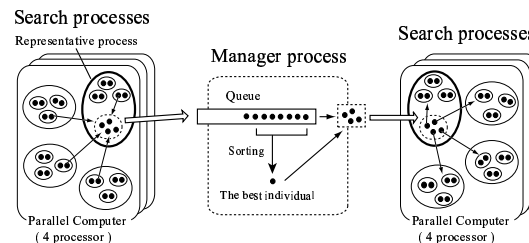


Fig. 3 Operation in the manager's queue

スタサイトと呼ぶ。その他のサイトはワーカサイトと呼び、それぞれ Dual DGA の並列モデルを実行する。つまり、検討モデルは Fig. 2 に示すようなマスタ・ワーカ型のモデルとなる。

検討モデルは次のように計算を行う。管理プロセスは、まずワーカサイトの情報を取得し、Dual DGA を実行するサイトを決定する。次に計算を実行するサイトに対して、Dual DGA が必要とするパラメータを送信し、Dual DGA（探索プロセス）を起動する。これは計算途中でも可能であり、必要に応じて動的に計算資源を追加し、計算に参加させることができる。

計算が始まると、管理プロセスはワーカサイトに対して能動的なチェックポイントを行なう。まずワーカサイトの探索プロセスと通信する。ただし通信のできない探索プロセスは無視する。そして、その時点での最良な探索点情報（GA での個体）を取得する。最良な探索点はワーカサイト内の並列プロセスから 1 つずつ集める。例えば、Dual DGA を 4 プロセッサで実行しているサイトからは、4 つの探索点情報を取得できる。取得した個体はキューに格納する。同時にキューから個体を取り出して、取得した個体数分の個体を探索プロセスに返送する。ただし、返送される個体にはその時点でキューに格納されている最良の個体を必ず含むようにする。Fig. 3 にキューの操作について示す。本研究ではキューに格納できる個体数は無制限としている。

この仕組みにより、ある探索プロセスの探索点情報が別の探索プロセスへ送られることになる。計算途中でサイトが追加されたり、あるサイトの計算がリセットされたりした場合には、キューに格納されている探索点情報が利用されることになる。

### 3.3 構築したシステム

本研究では Globus ツールキットを利用して 3.2 節のモデルを実現するシステムを構築した。システムは Manager, GASSサーバ, Searcherのコンポーネントからなり、ユーザがマスタサイトにおいて Manager を起動することで計算が始まる。起動された Manager は GRAM (Globus Resource Allocation Manager) を利用して、各ワーカサイトで GASSサーバと Searcher を起動する。Searcher は Dual DGA の実行プロセスであり、定期的に各島の最良探索点情報をファイルに書き出す。チェックポイントでは、Manager は GASS (Globus Access to Secondary Storage) を利用して、GASSサーバと通信し、Searcher の書き出したファイルを読み込む。そしてキュー操作を行なった後に、Searcher が読み込むべきファイルを書き出す。

## 4 数値実験

### 4.1 実験方法

本研究では 3.2 節で説明したモデルについて評価実験を行なった。実験は疑似 Grid 環境を用意し、以下に示す 3 種類のシナリオ下で行なった。

(シナリオ 1)

計算の最初から最後まで全てのサイトが問題なく稼働し、ネットワークも正常に利用できる。

(シナリオ 2)

計算の途中にマスタサイトとあるワーカサイト間のネットワークが途切れる。障害の間、管理プロセスはそのワーカサイトに対してチェックポイントを行うことができない。しかし、そのワーカサイトの探索プロセスはチェックポイントを待つことなく、オーバーラップして探索を行なう。本シナリオはタイムステップ毎に乱数を発生させて、ネットワークが利用できるかどうかを確率的に決定することで作成した。

(シナリオ 3)

計算の途中にワーカサイトが停止したり、復旧したりする。例えば、Fig. 4 に示すようなシナリオを用意した。本シナリオもシナリオ 2 と同様の方法で作成した。ただし、直前のタイムステップに停止していたサイトは次のタイムステップにおいても停止する確率が高くなるように設定した。

対象問題には 30 次元の Ridge 関数を最小化する問

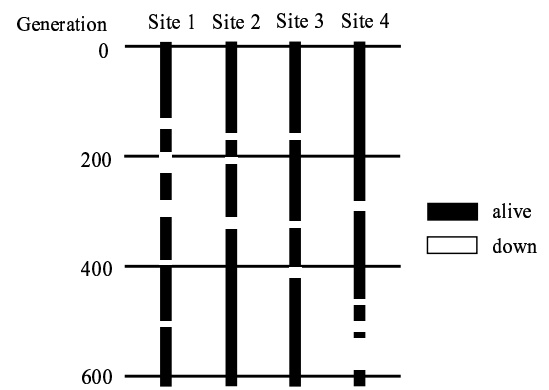


Fig. 4 Example of Scenario 3

Table 1 Parameter setting for Dual DGA

Number of islands	96
Number of processes	4
Coding method	10bit, graycode
Crossover method	1pt crossover
Mutation Rate	1/Length of the gene
Migration topology	Ring
Intramigration gap	5
Intermigration gap	Intramigration gap $\times$ 5
Intermigration rate	0.1 (1 island at least)

題を用いた。ただし、結果を対数グラフで表示できるように下記のように式を変更して実験を行った。この時、最適値は  $f = 1$  となる。ワーカサイトで実行する Dual DGA のパラメータには Table 1 を用いた。

$$f = \sum_{i=1}^n \left( \sum_{j=1}^i x_j \right)^2 + 1, \quad (-64 \leq x_i \leq 64) \quad (1)$$

### 4.2 シミュレーションによる実験

シミュレーション実験では 16 プロセッサの並列計算機を用い、これを仮想的に 4 分割して計算を行なった。この時、マスタサイトの機能はいずれかのプロセッサに受け持たせた。

(1) キュー操作と解探索性能

検討モデルのキュー操作が解探索の性能に及ぼす影響を確認するため、チェックポイント時にキュー操作を行なうモデルと行なわないモデルについて比較した。シナリオ 1 と 2 を用い、チェックポイント間隔を 100 世代として実験を行なった。Table 2 に 30 試行の平均結果を示す。これより、定期的にチェックポイントを行ない、かつキュー操作を施すモデルが最も早く最適解を得ているのが分かる。また、信頼性の低いネットワークを用いた場合でもチェックポイント機能を用いることで、より早く最適解を見つけている。

Table 2 Results of the scenario 1 and 2

	SC1	SC1	SC2
Queue operation	yes	no	yes
GR	1971	2284	1949
DR	0.0	-	0.16

SC: Scenario

GR: Generation where the optimum was found

DR: Rate of the network which is disconnected

Table 3 Results of the scenario 3

Queue operation	yse	yes	no
CP Interval	50	100	-
RO	1.0	0.87	0.0
UR	0.61	0.41	-

CP: Check point

RO: Rate of reaching the optimum

UR: Rate of the usable resource

## (2) ワークサイトの停止による影響

シナリオ 3 において、キュー操作を行なうモデルと行なわないモデルについて比較した。チェックポイント間隔を 100 世代とした時の結果の一例を Fig. 5 に、30 試行の結果を Table 3 に示す。UR は全リソース (リソース = サイト数 × 世代数) のうち、計算に使うことのできたリソースの割合を示している。これによると、チェックポイント間隔を短くしてキュー操作を行なうモデルが最もよく最適解を発見できている。また、リソースが多く確保できれば、解を発見できる確率が高くなるという結果も示されている。

## 4.3 システムの評価実験

構築したシステムを 10BASE-TX のイーサネットに接続された 2 つの並列計算機に導入し、チェックポイントに要する時間を測定した。結果として両サイトで実行する探索プロセス数を増加させても計測時間の増加は見られず、約 1.5 秒程度であった。つまり、この時間内に GA の進化が何世代も進んでしまうことのない、ある程度の規模をもつ最適化問題であれば本システムは十分に有効であると思われる。

## 5 おわりに

本論文では、広域に存在する複数の計算資源を利用できる Grid 計算環境を想定し、その環境に適応した GA のモデルについて検討を行った。シミュレーション実験を通じて、検討モデルが Grid 環境を利用して高速に解探索が可能であり、Grid 環境で起こりうるであろう障害に対してロバストなモデルであることが確認できた。そしてシステムの構築とその評

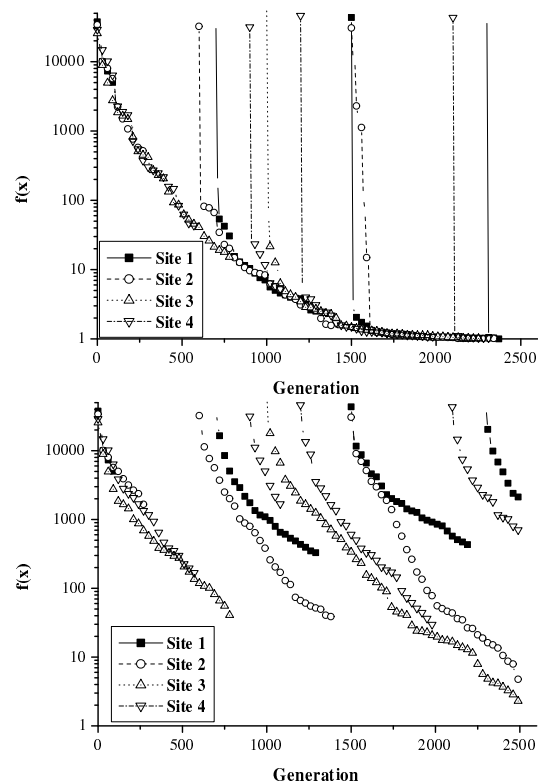


Fig. 5 Result of the scenario 3 – an example  
(up: queue operation, down: no queue operation)

価実験により、検討モデルが実現可能なモデルであることが確認できた。

謝辞 本研究は文部科学省科学研究費補助金、および文部科学省学術フロンティア推進事業により支援されている。

## 参考文献

- 1) Enrique Alba and Jose M. Troya. A Survey of Parallel Distributed Genetic Algorithms. *Complexity*, Vol. 4, No. 4, pp. 10–11, 1999.
- 2) Henri Casanova and Jack Dongarra. Netsolve: A Network Server for Solving Computational Science Problems. *The International Journal of Supercomputer Applications and High Performance Computing*, Vol. 11, No. 3, pp. 212–223, 1997.
- 3) D. E. Goldberg. Genetic algorithms in search. *optimization and machine learning*, 1989.
- 4) Tomoyuki Hiroyasu, Mitunori Miki, Masahiro Hamasaki, and Yusuke Tanimura. A New Model of Distributed Genetic Algorithm for Cluster Systems: Dual Individual DGA. In *Proceedings of CC-TEA*, Vol. 1, pp. 477–483, 2000.
- 5) 谷村勇輔, 廣安知之, 三木光範. PC クラスタにおける 2 個体遺伝的アルゴリズムの高速化. 情報処理学会研究報告 HPC 研究会, Vol. 2000, No. 73, pp. 161–166, 2000.

## 6 出展

日本機械学会 No.01-35

第 11 回設計工学・システム部門講演会講演論文集

2001/11/5-7, 香川

ページ 126-129