

# グローバルコンピューティング環境における 遺伝的アルゴリズムのモデルの検討

谷村勇輔，廣安 知之，三木光範，佐野正樹

同志社大学 工学部 知識工学科

E-mail: tomo@is.doshisha.ac.jp

本研究では，グローバルコンピューティング環境における遺伝的アルゴリズム（GA）のモデルの検討を行い，仮想的環境におけるシミュレーションを行った．使用モデルはマスター・スレーブモデルである．マスターサイトがスレーブサイトの計算状況をチェックする際に各サイトでの最良探索点を収集し，マスターサイトの持つキューに確保する．また同時に，キューに確保されている他サイトの探索点情報を送信することで，サイト間の情報交換を実現している．検討の結果，使用モデルは，グローバルコンピューティング環境で発生するサイトの緊急停止や大きな通信遅延に対応可能であることが明らかとなった．

## Genetic Algorithms in a Global Computing Environment

Yusuke TANIMURA, Tomoyuki HIROYASU, Mitsunori MIKI, and Masahiro SANŌ

Knowledge Engineering Department, Doshisha University

In this study, a model of Genetic Algorithms (GA) under the global computing environment is discussed through the numerical experiments. The proposed system is a master slave model. In this system, a master site reserves the information of the best searching point of a slave site in a queue when the master checks the calculation condition of the slaves. At the same time, the master site sends the other information of the searching point that is derived in the other slave sites to the other slave sites. By this operation, the searching information is transferred between the slave sites. The proposed model can solve the problems in the global computing environment; the sudden halt of the site and the huge network latency. These characteristics of the proposed model are confirmed and discussed through the numerical experiments.

### 1 緒言

1950 年頃より始まったノイマン型デジタルコンピュータは目覚ましい進歩を遂げ，現在の科学技術の研究に必要不可欠なものとなっている．様々な機関は大規模な計算を行うため，スーパーコンピュータと呼ばれる超高性能の計算機を導入している．さらに，1990 年代に入りインターネットの普及と共にネットワーク環境が整備されてきた．そこで 1 つの機関だけで計算資源を利用するのではなく，広い地域に配置された計算資源やその他の資源を結びつけ，広域的に分散 / 並列計算を行うグローバルコンピューティングと呼ばれる新しい計算モデルが研究されるようになってきた．<sup>1) 2)</sup>

現在はテストベッドを構築・維持するという形で，欧米を中心にグローバルコンピューティングの研究が推し進められている．そして Globus<sup>4)</sup> や Legion<sup>5)</sup> といったグローバルコンピューティング

環境の基本システムが開発されている．このようにグローバルコンピューティングを実現する環境は整いつつある．しかし一方では，そうした環境をどのように実際の計算へ応用するのか，またそこで何ができるのかといった応用分野の研究は H.Casanova らの NetSolve の研究<sup>6)</sup> や佐藤らの netCFD の研究<sup>7)</sup> などがあるが，ほとんど行われていない．

そこで本研究では，アプリケーションとして最適化に着目する．最適化は構造物設計，タンパク質の構造解析などの際に利用される技術で，解析を繰り返し行うために非常に計算コストが高くなる．最適化アルゴリズムはいくつか存在するが，遺伝的アルゴリズム（Genetic Algorithms : GA）を取り上げ，GA がグローバルコンピューティング環境における最適化に適していることを検討する．

ある目的関数の値を制約条件内で最小化もしくは最大化する設計変数を決定する問題は，最適化

問題と呼ばれる。最適化問題を数値計算により処理するためには、現探索点情報から次探索点を決定する最適器と、新探索点からその制約条件および目的関数の値を決定する解析器を設計し、それらを繰り返し利用することが必要となる。後で詳しく説明する通り、グローバルコンピューティング環境においては、資源の増減が動的に行われるため、最適器と解析器の繰り返しが途切れてしまい、解探索が行えない可能性が生じる。そのため、グローバルコンピューティング環境において最適化を行うためには、この問題を克服するモデルが必要となる。

遺伝的アルゴリズム（以下 GA）は最適化手法の 1 つであり、生物の遺伝と進化を模擬したアルゴリズムである。<sup>8)</sup> GA は多点探索手法であり、高い並列性を有している。同時に各探索点がお互いに進化し淘汰して解を探索していくために、たとえば探索点の一部が失われたとしても、それらの情報が陰に残りの探索点に残っているものと考えられる。このような特徴は、グローバルコンピューティング環境において最適化を行うのに適した手法であるといえる。そこで本研究では、グローバルコンピューティング環境において最適化を行う際の GA の有効性を検討する。

使用した GA は、筆者らが提案している 2 個体分散遺伝的アルゴリズム（Dual Individual Distributed Genetic Algorithms : Dual DGA）<sup>9)</sup> を、グローバルコンピューティング環境に拡張したモデルである。

使用したモデルは、ハイパークラスタに対応した Dual DGA のマスター・スレーブモデルである。本論文では、ハイパークラスタを構成する各クラスタをサイトと呼ぶ。マスターサイトは、スレーブサイトに対して Dual DGA の各パラメータを与え Dual DGA を起動する。各スレーブサイトでは通常の Dual DGA の計算を行い、探索を行う。マスターサイトは、一定時間間隔でスレーブサイトの状態をチェックするが、その際に、各スレーブサイトの最良探索点の状態を取得する。そして、マスターサイトは他サイトのチェック時に別サイトの探索点を送信する。この操作により、最適解は高速に求まることになる。同時に、あるスレーブサイトが停止し、それまでの計算が強制的に終了されてしまっても、マスターサイトが取得している他サイトの探索情報を利用し再開することで、効

率的な計算が期待できる。これは、グローバルコンピューティング環境において予想されるネットワークの遅延や過負荷、スレーブサイトの不慮の停止などの障害に対して、ロバストなモデルであるといえる。

本研究ではクラスタシステム上で擬似グローバルコンピューティング環境を構築し、いくつかのシナリオを想定し検討を行っている。

## 2 グローバルコンピューティング環境

グローバルコンピューティング環境は、従来のコンピューティング環境に比べて次のような特徴を有する。<sup>10)</sup>

- 1) 計算は常にネットワークを介して行われる。
- 2) 計算には非常に多くの資源が関与し、その構成は動的に変化する。
- 3) 多数の組織から資源が提供される。
- 4) 計算は複数の資源にまたがって行われる。

これはグローバルコンピューティングで動作するアプリケーション開発者にとって、次のように言い換えられる。

遠隔地の計算拠点間で計算を行う場合、それをつなぐネットワークの構成により、高速にアクセスできる拠点とそうでない拠点が存在してしまう（特徴 1）。様々な障害により、ネットワークに大きな遅延が発生したり、完全に切断されたりする可能性がある（特徴 2）。計算拠点自体が障害やメンテナンスのために停止する可能性がある（特徴 3）。各計算拠点は等しい計算資源を持っているわけではない。中には高速な計算機を有している計算拠点もあれば、そうでない計算拠点もある。別の計算拠点は大規模なストレージを有しているかもしれない。つまり、グローバルコンピューティング環境は多くの場合、非均質な計算機環境から構成されることになる（特徴 4）。ユーザは上記の制約のもと、計算資源を有効に利用しなければならない（特徴 5）。

## 3 2 個体分散遺伝的アルゴリズム

遺伝的アルゴリズム（Genetic Algorithms : GA）は優れた最適化手法の 1 つである。<sup>8)</sup> GA は生物の進化と淘汰を模倣した確率的な多点探索を行う。各探索点は個体と呼ばれ、個体の集合は母集団と呼ばれる。GA では、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，古い個体は親個体，新しい個体は子個体と呼ばれる．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ことができる．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．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．

分散 GA は島モデルとも呼ばれる．並列 GA モデルの 1 つである島モデルは他の GA モデルと比較して解探索能力に優れ，分散メモリ型並列計算機にも実装しやすいという特長をもつ．<sup>12) 13)</sup>

2 個体分散遺伝的アルゴリズム (Dual Individual Distributed Genetic Algorithms: Dual DGA) は，島モデルを拡張したモデルである．<sup>9)</sup>

Dual DGA では，島内の個体数を 2 と設定する．これにより，島数，交叉率，移住率などのいくつかのパラメータが一意に決まり，設定すべきパラメータ数が減少する．Dual DGA では，できるだけ全体の多様性が維持できるよう，交叉や突然変異の方法を従来のものから変更している．

Dual DGA の特徴の 1 つは高い探索能力である．それを示すために，次式で示す Ridge 関数の最小化を行う．実験で用いた Ridge 関数の次元数は  $30(n = 30)$  としている．

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

用いたパラメータは表 1 に示す通りである．図 1 に Dual DGA と SPGA の評価回数に対する最良の適合度値の変遷を示す．これらは 20 回試行の平均である．

表 1: Parameter setting for Dual DGA

Number of islands	192
Number of processes (in parallel)	4
Coding method	10 bit, graycode
Crossover method	1pt crossover
mutation rate	1 / Length of the gene
Migration topology (intra, inter)	Ring
Intra migration gap	5
Inter migration gap	Intra migration gap × 5
Intrr migration rate	0.1 (1 islands at least)

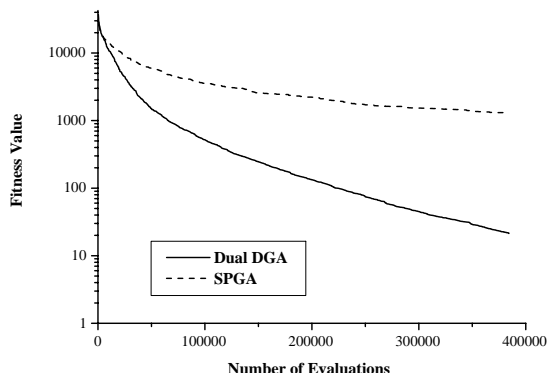


図 1: SPGA v.s. Dual DGA

この図からも分かる通り，Dual DGA は SPGA に対して少ない評価回数で解が求まる．この傾向は，GA が解探索困難な問題以外は，いくつかのテスト関数に対して同様である．

Dual DGA をクラスタのような分散メモリ型並列計算機に実装する際には，次のようにモデルを変更する．すなわち，1 つのプロセッサに対して複数の島を割り当て，各プロセッサ毎に Dual DGA を実行する．ただし，プロセッサ間で逐次モデルと同じような移住 (プロセス間移住) を行う．この移住では個体の交換を行うのではなく島の交換を行う．そしてプロセス間移住は，プロセス内移住に比べて大きな移住間隔をとる．図 2 に移住の全体図を示す．このような移住モデルを採用することにより，通信負荷を減らすことができる．

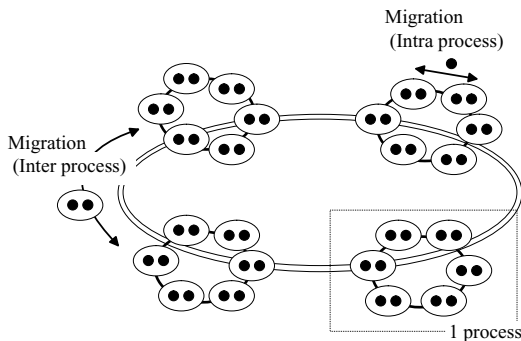


図 2: Migration model of Dual DGA

Dual DGA の利点は，並列計算機に実装した際の通信負荷が減少可能な点である．また，Dual DGA の島数とプロセッサ数が 1 対多の関係になるために，並列化の粒度を柔軟に変更できる点が優れている．これは，実行する並列計算機の (通信 / 計

算)の割合に応じた粒度を用いることができたり、非均質な並列計算機においてロードバランスがとりやすいということを意味する。これらの利点は、今回想定するような遠隔地にあるクラスタを多数利用してグローバルコンピューティングを行おうとする場合に、非常に有効に働くと考えられる。

Dual DGA とその並列モデルの特徴をまとめると以下ようになる。

[Dual DGA]

- 分散 GA の仕組みを受け継いでいるが、分散 GA に比べてアルゴリズムが簡易化されている。
- 従来のも単純 GA や分散 GA に比べて、解探索能力が高い。

[Parallel Dual DGA]

- 高速に計算を行える。
- 通信量が少ない
- 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。

4 グローバルコンピューティング環境における Dual DGA モデル

本研究では、グローバルコンピューティング環境における GA の検討を行う。使用したモデルでは Dual DGA のモデルを用い、複数の計算拠点を用いて並列に探索を行う。本研究では計算拠点として、クラスタのような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。これをマスターサイトと呼ぶ。マスターサイト以外のサイトはスレーブサイトと呼び、それぞれ Dual DGA の並列モデルを実行する。使用モデルでは、マスターサイトは計算の最初から最後まで停止しないと仮定している。つまり使用モデルは図3に示すように計算が行われることになり、マスター・スレーブモデルであるといえる。

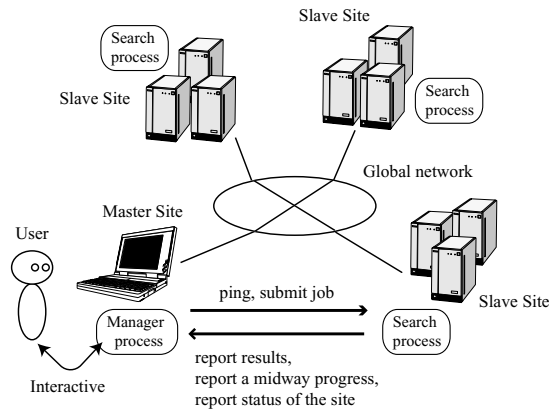


図 3: Concept of proposed model

マスターサイトは以下の機能を有するが、(1) や (2) の機能は Globus などのツールキットを利用して実装が図られるべきであると考え、本論文ではこれらの詳細については考えない。

- (1) スレーブサイトの計算資源に関する情報(計算負荷、利用の可否)を取得する。
- (2) スレーブサイトに対して Dual DGA の実行ジョブを投入する。
- (3) スレーブサイトとの間において探索点情報をやり取りする。
- (4) 探索点情報を格納するためのキューを保持する。

使用モデルは次のように計算を行う。

管理プロセスは、まず最初にスレーブサイトの計算資源に関する情報を取得し(1)、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(2)。この作業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も可能である。追加する計算資源はサイト単位で行う。

計算途中では、ある一定時間毎にチェックポイントを設ける。チェックポイントにおいて、管理プロセスはスレーブサイトに関する情報を取得する(1)。計算を続行できる場合にはさらに探索プロセスと通信し、その時点での最良な探索点情報(GA のの個体)を取得する(3)。最良な探索点はスレーブ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(4)。同時にキューから個体を取り出して、取得した個体数だけの個

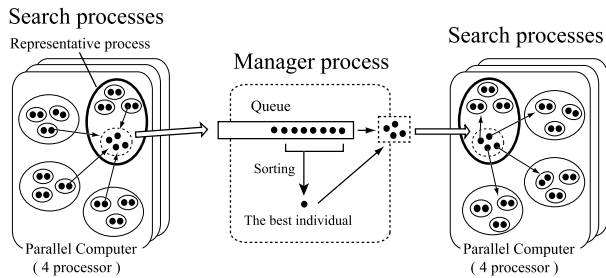


図 4: Queuing operation

体を探索プロセスに返送する (3) . 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. 図 4 にキューの操作について示す. また, 本研究ではキューに格納できる個体数は最大 32 個体としている.

この仕組みにより, ある探索プロセスにおける探索点情報が別の探索プロセスへ送られることになる. また, 計算途中に新たに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. 一方, チェックポイント後に計算が続行できない探索プロセスは, 探索点情報を管理プロセスに送るだけで, 管理プロセスから探索点情報を受け取ることはない. また, 応答のなくなったスレーブサイトに対しては何も操作を行わない.

## 5 数値実験

### 5.1 検討するシナリオ

本研究では GA をシミュレーションとして実装し, その評価を行った. 想定するグローバルコンピューティング環境は 1 つのマスターサイトと 4 つのスレーブサイトから成る. 各スレーブサイトでは 4 ノードの並列計算機が稼動していると仮定している. 実験では, グローバルコンピューティング環境をシミュレートするために, サイト単位で計算を再開・停止させたりネットワークが過負荷になるような状況を作り出したりした. このような障害に関するシナリオはあらかじめ用意しておき, 各シナリオに対して計算を行った. 本研究で用意したシナリオは以下の 3 種である.

#### (Scenario 1)

計算の最初から最後まで 4 つのスレーブサイトが問題なく稼動し, 計算途中に環境が変化しない.

本シナリオは, 何も障害が起きない場合に使用モデルが有効であるかを検討するために用意した.

#### (Scenario 2)

計算途中にマスターサイトとスレーブサイト間のネットワークが, 不規則的に過負荷となる.

使用モデルでは, このような障害が生じた場合, 管理プロセスが探索プロセスと通信を行うことができないために, チェックポイントすることができない. しかし計算を効率良く行うためには, 探索プロセスは管理プロセスとの通信を待つことなくオーバーラップして計算を行う必要があると思われる. そこで, 提案モデルにその手法を実装し, 本シナリオを試すことでその有効性を検討する.

本シナリオの作成では, チェックポイント毎に管理プロセスからのロックングに対して, 応答するかどうかということを任意の確率を用いて決定している.

#### (Scenario 3)

計算途中にスレーブサイトが停止したり, 再び稼動したりする. 停止したサイトでは探索プロセスが強制終了されるために, それまでの探索点情報が失われる. 本シナリオの資源利用の状態を世代数で表した例を図 5 に示す.

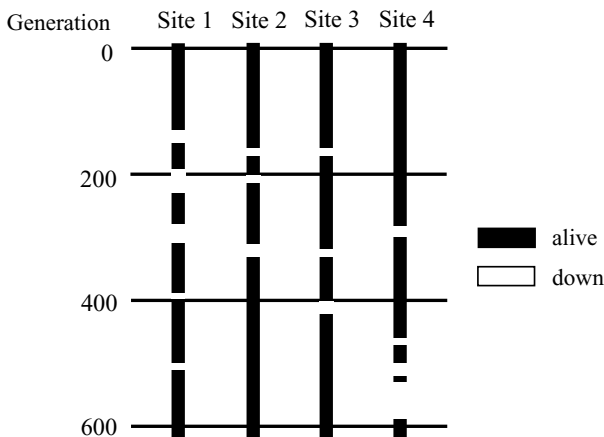


図 5: Example of Scenario 3

提案モデルでは, チェックポイントとキュー操作の仕組みがあるために, このような障害が生じてもその影響を抑えることができるとされる. そこで, 本シナリオを試すことで提案モデルの有効性を検討する.

本シナリオの作成では 10 世代毎にタイムステップを区切り、タイムステップ単位で走行・停止を決める。サイトが停止する確率、および停止するタイムステップ数には任意の確率を用いる。

## 5.2 対象問題とパラメータ

対象問題には 30 次元の Ridge 関数を最小化する問題を用いる。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。Dual DGA を行う時の各種パラメータは、表 1 のものを用いる。

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

## 5.3 数値実験結果

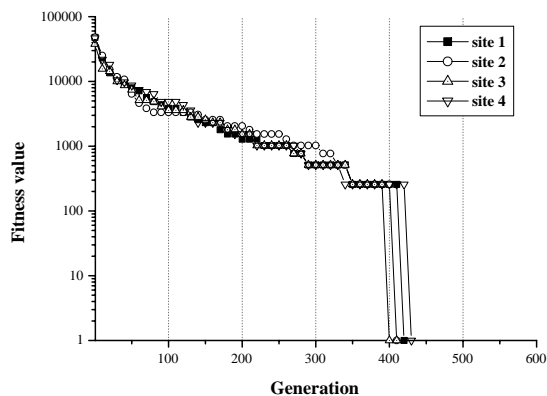
### 5.3.1 キュー操作と解探索性能

提案モデルにおけるキュー操作が解探索の性能に及ぼす影響について検討を行った。本実験では、チェックポイントにおいてキュー操作を行う提案モデルとチェックポイントにおいてキュー操作を行わないモデルについて比較した。本実験は (Scenario 1) と (Scenario 2) を用いた。(Scenario 1) ではチェックポイント間隔を 100 世代とし実験を行い、(Scenario 2) ではチェックポイント間隔を 30 世代として実験を行った。結果の一例を図 6 に示す。表 2 には、それぞれ 10 試行の結果の平均を示す。表中の SC はシナリオ番号を表す。同様に GR は最適解発見までの世代数を表し、RC はリソース消費量 (計算を行った世代数  $\times$  サイト数) を比率で表す。また、ER はチェックポイントにおいてキュー操作が行われた比率を表す。

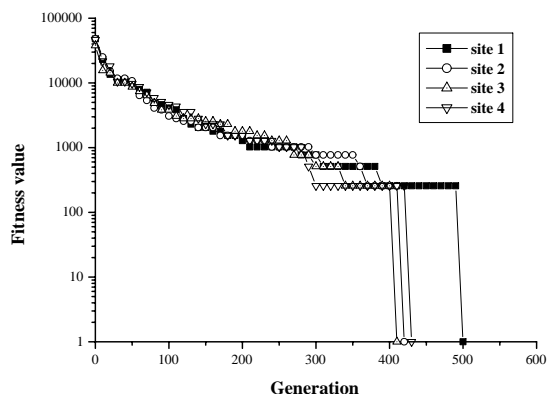
これらの結果より、(Scenario 1) において提案モデルの方がキュー操作を行わないモデルに比べて短い世代数で最適解を発見でき、リソースの消費量が少なく済むのが分かる。さらに、(Scenario 2) のようにネットワークに障害が生じる場合でも、できる限りキュー操作を行った方が良いことが分かる。

表 2: Effectiveness of the queue system

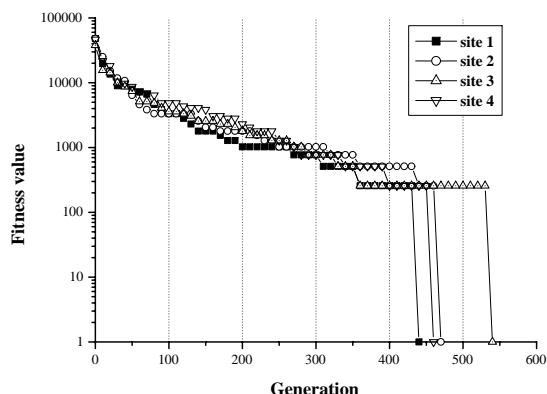
	SC 1	SC 2	SC 1
Queuing	yes	yes	no
GR	328.7	378.0	400.4
RC	1.00	1.15	1.22
ER	1.00	0.60	0.00



a) Scenario 1 (queuing)



b) Scenario 2 (queuing)



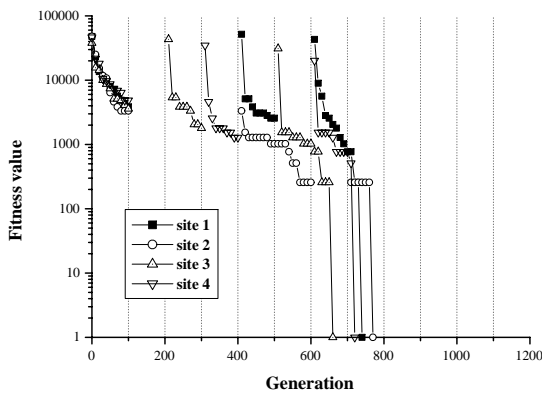
c) Scenario 1 (no queuing)

図 6: Queuing effects

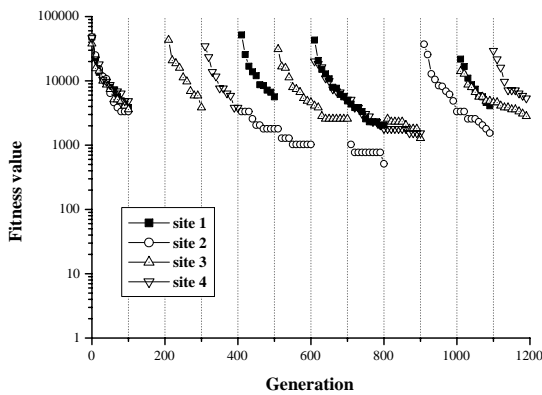
### 5.3.2 キュー操作とチェックポイントニング

提案モデルにおけるキュー操作とチェックポイント間隔について検討を行った。まず初めに、(Scenario 3) においてキュー操作を行う提案モデルと

キュー操作を行わないモデルについて比較を行った。このシナリオでは、サイト停止により探索点情報が途中で部分的に失われる。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、図 7 に示す。図 7 のように、キュー操作を行わないモデルは 1200 世代以内に最適解を発見できない場合が多かった。表 4 はそれぞれ 30 試行の結果である。これによると、提案モデルではチェックポイント間隔に関わらず最適解に到達することができているが、キュー操作を行わないモデルは最適解を発見することができなかった。表中の CPTI はチェックポイント間隔を表し、提案モデルでは 30 世代、50 世代、100 世代と変化させている。



a) Queuing



b) No queuing

図 7: Example of Scenario 3

次に、チェックポイント間隔を 30 世代、50 世代、100 世代とした時の解探索の性能やリソース消費量について検討を行った。表 4 に各 10 試行の結果

を示す。表中の CPTI はチェックポイント間隔を表し、GR は最適解が見つかるまでの世代数を表し、RC は最適解が見つかるまでのリソース消費量（計算を行った世代数×サイト数）を比率で表している。ER は、スレーブサイトが有効に計算を行えた世代数の割合を示している。有効な計算とは、あるチェックポイントから次のチェックポイントまで停止することなく行うことのできた計算のことである。

表 3: Rate of reaching Opt.

Queuing	yse	yes	yes	no
CPTI	30	50	100	-
解発見率	1.0	1.0	0.7	0.1

これらの結果より、チェックポイント間隔を短くした方がスレーブサイトの停止をより早く検知できるため、ER を上げることができる。そして ER が高いほど、最適解を見つけるまでの世代数が短くなるのが分かる。一方、リソース消費量はチェックポイント間隔にあまり依存していないという結果が得られている。

表 4: Checkpointing Interval

CPTI	30	50	100
GR	471	592	721
RC	1.00	1.04	0.97
ER	0.76	0.63	0.48

## 5.4 考察

キュー操作はチェックポイントの前後において、GA の個体をやり取りする操作である。やり取りされる個体は、主に適合度の高い個体である。GA では、適合度の高い個体ほど多くの探索点情報を含んでいる確率が高い。つまり、提案モデルはそのような個体を媒介として探索プロセス間の情報交換を行うモデルである。これは、分散 GA における移住と同じような仕組みを提供し、各探索プロセスが高い多様性を維持して探索を行うことを可能にしている。結果として、5.3.1 節の実験結果が示しているように、提案モデルは拘束に解探索を行うことができる。さらに、(Scenario 2) のような信頼性の低いネットワークでも有効に働くモデルであるといえる。

一方、チェックポイントとキュー操作の仕組みは、キューに個体を保持することで計算の途

中経過のバックアップを行っている。これにより、障害により探索情報が失われたり探索が遅れていたりするスレーブサイトや、新たに計算に加わるスレーブサイトが、その時点での最良の探索点をもとに速やかに探索を開始できる。このような仕組みを用いることができるのは、GA が多点探索で比較的独立性の高い探索を行うためである。

またチェックポイント間隔は通信コストが許す限り、頻繁に行うのが良いという結果が得られた。

## 6 結論

本論文では、遠隔地に存在する複数の計算資源を利用するグローバルコンピューティング環境を想定し、その環境に適応した Genetic Algorithms (GA) のモデルについて検討を行った。GA としては、高い探索性能をもち並列モデルとしても優れている Dual DGA を用いた。我々が使用したモデルはマスター・スレーブモデルであり、キュー操作の仕組みを備えたチェックポイント機能をもっている。仮想的なグローバルコンピューティング環境におけるシミュレーションを通じて、今回使用した GA モデルが以下の特徴を有するモデルであることを確認した。

- 1) 複数の計算資源を利用することで、より高速に解探索が行える。
- 2) スレーブサイトが緊急に停止したり、ネットワークが途切れたりといった障害に対してロバストである。
- 3) 計算途中で資源を追加した場合にでも、それを有効に利用できる。

グローバルコンピューティング環境において最適化を行う際に、計算資源の増減やネットワークの速度は、解探索情報の消失や資源利用の無駄につながるために大きな問題となる。上記の特徴により、使用モデルはグローバルコンピューティング環境における問題点を克服することができ、効率的に解探索が可能であることが明らかとなった。

## 参考文献

- 1) Ian Foster, Carl Kesselman, "The Grid: Blueprint for a New Computing Infrastructure", Morgan Kaufmann, 1998
- 2) 関口智嗣, "グローバルコンピューティングとは", Computer Today, No.95, 2000
- 3) 関口智嗣, "グローバルコンピューティングテクノロジー", Computer Today, No.96, 2000

- 4) Ian Foster, Carl Kesselman, "Globus: A meta-computing infrastructure toolkit", International Journal of Supercomputing Applications, 1997
- 5) A.S. Grimshaw, W.A. Wulf, the Legion team, "The Legion vision of a worldwide virtual computer", Communications of the ACM, 1997
- 6) Henri Casanova, Jack Dongarra, "NetSolve: A Network Server for Solving Computational Science Problems", The International Journal of Supercomputer Applications and High Performance Computing, Volume 11, 1997
- 7) Mitsuhiisa Sato, Kazuhiro Kusano, Hidemoto Nakada, Satoshi Sekiguchi, Satoshi Matsuoka, "netCFD: a Ninf CFD component for Global Computing, and its Java applet GUI", Proceedings of the 4th HPC Asia, 2000
- 8) D.E. Goldberg, "Genetic algorithms in search", optimization and machine learning, Addison-Wesley, 1989
- 9) Tomoyuki Hiroyasu, Mitsunori Miki, Masahiro Hamasaki and Yusuke Tanimura, "A New Model of Distributed Genetic Algorithm for Cluster Systems: Dual Individual DGA", Proceedings of CC-TEA, 2000
- 10) 中田秀基, "グローバルコンピューティングにおけるセキュリティ", Computer Today, No.99, 2000
- 11) Enrique Alba, Jose M. Troya, "A Survey of Parallel Distributed Genetic Algorithms", Complexity Vol.4 No.4, John Wiley & Sons, Inc., 1999
- 12) Reiko Tanese, "Parallel Genetic Algorithms for A Hypercube", Proceedings. of 2nd ICGA, 1987
- 13) Tomoyuki Hiroyasu, Mitsunori Miki, Yusuke Tanimura, "The Differences of Parallel Efficiency between the Two Models of Parallel Genetic Algorithms on PC Cluster Systems", Proceedings of the 4th HPC Asia, 2000
- 14) Marco Dorigo, Vittorio Maniezzo, "Parallel genetic algorithms: Introduction and overview of current research", Parallel Genetic Algorithms: Theory and Applications, IOS Press, 1993
- 15) Hidemoto Nakada, Mitsuhiisa Sato, Satoshi Sekiguchi, "Design and Implementations of Ninf: towards a Global Computing Infrastructure", Future Generation Computing Systems, Metacomputing Issue, 1999
- 16) 鈴村豊太郎, 中川貴之, 松岡聡, 中田秀基, "クライアント・サーバ型のグローバルコンピューティングシステムの比較 - Ninf, NetSolve, CORBA, Ninf-on-Globus の性能評価 - ", 情報処理学会研究報告, 99-HPC-34, 1999
- 17) 田中良夫, 平野基孝, 佐藤三久, 中田秀基, 関口智嗣, "Firewall に対応した Globus による広域クラスタシステムの構築と性能評価", 情報処理学会システムハイパフォーマンスコンピューティング研究会, vol.2000, No.57, 2000
- 18) Rajkumar Buyya, "High Performance Cluster Computing, Vol.2, Programming and Applications", Prentice Hall, 1999



- 出典

並列処理シンポジウム JSP2001

2001年6月5 - 8日, 京都リサーチパーク

予稿集 p.279 - 286