

Dual DGA のグリッドモデル

谷村勇輔

同志社大学大学院 工学研究科

E-mail: tanisuke@mikilab.doshisha.ac.jp

近年のコンピュータ技術の進歩とネットワークの発達には目を見張るものがある．これらを利用して、遠隔地に点在する複数の計算資源を結びつけて計算を行うという新しい計算モデルが提案されている．これらは一般にグローバルコンピューティング、あるいはグリッドと呼ばれている．現在、このグローバルコンピューティング環境を構築するための基本的なソフトウェアが多数研究され、かなり実用的なレベルに近づきつつある．しかし、実際にこの環境を利用したアプリケーションは数少ない．本論文は、グローバルコンピューティング環境を利用した遺伝的アルゴリズムのモデルを提案し、その検討を行った．そして提案モデルが、グローバルコンピューティング環境を有効に利用できることを確認した．

The Grid-enabled Model of the Dual DGA

Yusuke TANIMURA

The Graduate School of Doshisha University

These days, progress of the computer technology and networks is remarkable. Using them, a new model is supposed that we compute with some computing resources where are far away. It is generally called Global Computing or Grid. Now, there are a lot of research for the fundamental software of it. And, they are likely to close a practical level. But, there are few research for application using the Grid. In this paper, we supposed the model of Genetic Algorithms using the Grid. We examined this model and confirmed that it use the Grid.

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な不可欠なものとなっている．大規模な計算を行うために、様々な機関がスーパーコンピュータと呼ばれるような超高性能の計算機を導入している．しかし、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使うのではなく、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．

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 や Legion 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あまりなされていないと思われる．こうした背景より、我々はグローバルコンピューティング環境を利用したアプリケーションを考えてみた．

アプリケーションとしては、遺伝的アルゴリズム (GA) を用いる．GA は進化的な最適化手法であるが、多くの繰り返し計算を必要とすることから、概して計算負荷が高くなる．高い計算負荷を高速に処理するために、従来より分散／並列モデルの研究が推し進められている．また GA は高い並列性をもっていることから、グローバルコンピューティングにも比較的組み込みやすいアプリケーションであると考えられる．本論文では、GA をグローバルコンピューティング環境で計算するためのモデルを提案し、その評価を行う．

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がある．

- 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。

1) についていえば、遠隔地の計算拠点 (grid) 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。2) についていえば、様々な障害により拠点までのネットワークが途切れたり、拠点自体が障害やメンテナンスのために停止したりする。3) についていえば、多くの計算拠点では1つのジョブの実行に対して時間制限が設けられている。そうした制約の中で利用できる計算資源を有効に利用しなければならない。4) についていえば、複数の計算資源は等しい性能をもっているわけではない。中には飛びぬけて速い計算機をもっているところもある。すなわち、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。

グローバルコンピューティング環境における計算拠点は様々である。高性能な単一プロセッサのマシンであっても良いし、大規模な並列計算機であっても良い。しかし、我々が対象とする計算拠点は、PC クラスタ型の並列計算機である。PC クラスタは、汎用的な PC を多数ネットワークに接続して作られる並列計算機である。近年のコンピュータ技術の進歩により、汎用的な部品でも高い性能をもつようになってきている。しかも汎用的な部品は価格が低い。これらを用いて作られる PC クラスタは、非常に高いコストパフォーマンスを示すといわれている。このような利点から、今後、多くの計算資源を必要とするところにおいて、次第に PC クラスタが普及していくものと思われる。

3 Dual DGA のクラスタモデル

3.1 モデルの説明

本論文では、グローバルコンピューティング環境における GA のモデルを提案する。

GA は優れた最適化手法の1つである。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、

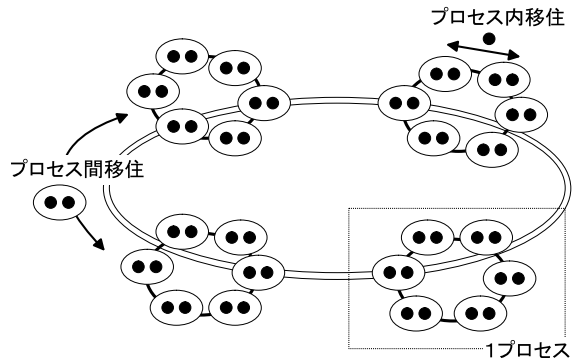


図 1: Parallel Dual DGA の移住モデル

新しい個体を生成する。この時点において、古い方の個体は親個体、新しい方の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。

一方、GA は潜在的な並列性をもっている。そこでその並列性を利用してより理想的な解をより高速に見つけることを目標にして、多くの並列モデルの研究がなされている。GA の並列モデルの中で、Dual DGA はクラスタ型の並列計算機に対して非常に有効である¹⁾。Dual DGA は母集団を2個体からなる多数の島に分割して、それぞれの島で独自に GA を行う。ただし、ある世代間隔において移住と呼ばれる個体交換の操作を行う。Dual DGA は、他の GA の手法に比べて多様性をうまく維持することができるため、良好な解を見つけることができる。

Dual DGA をクラスタ型の並列計算機に実装する時には、次のようにモデルを変更する。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 1 に、移住の全体図を示す。

3.2 数値実験

Dual DGA は SGA に比べて、効率よく解の探索が行えることが分かっている。図 2 に示す数値

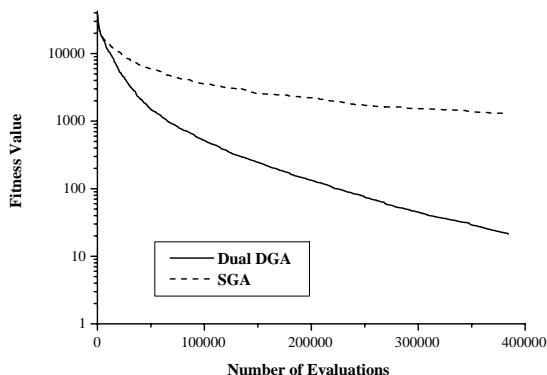


図 2: SGA v.s. Dual DGA

実験では，Dual DGA と SGA の比較を行った結果である．対象問題としては，代表的な最適化問題の 1 つである Ridge 関数を最小化する問題を用いる．Ridge 関数は以下の式で表され，今回の実験では 30 次元 ($n = 30$) の問題を解く．

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

3.3 Dual DGA の利点

Dual DGA の利点は，並列計算機に実装した時の通信負荷を小さくできることである．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．

4 Dual DGA のグリッドモデル

4.1 モデルの説明

3 章で述べた理由より，我々は Dual DGA をグローバルコンピューティング環境で実行できるように拡張した．このモデルでは，1 つの計算拠点がそれぞれ 1 つの Dual DGA，あるいは Parallel Dual DGA を行う．ただし，ある 1 つの計算拠点は他の拠点とコミュニケーションを行うための処理（管理プロセス）を実行することとし，その計算拠点は計算の最初から最後まで停止しないと仮定した．すなわち，計算拠点全体で見ると，図 3 のように計算が行われることになる．

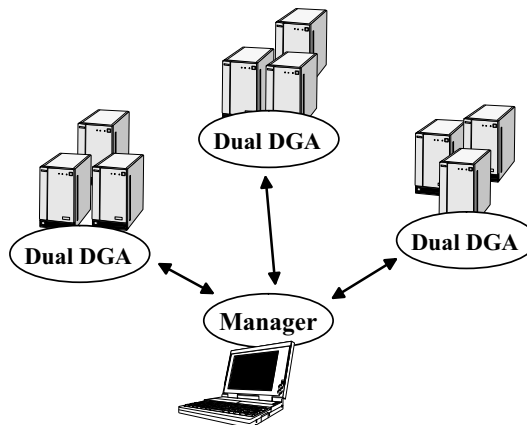


図 3: The Grid model of the Dual DGA

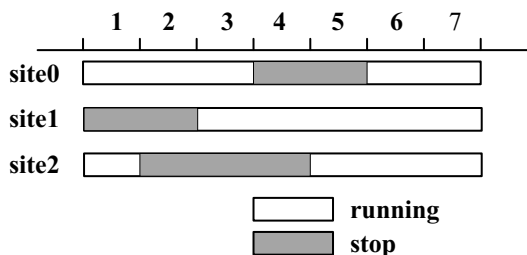


図 4: The plan of the site running

Dual DGA を実行する計算拠点では，十分な世代毎にチェックポイントを設けた．このチェックポイントとチェックポイントの間を 1 クール (?) という単位で考えて，チェックポイント時に次の 1 クールの間，この拠点の計算資源が利用できるかどうかを決定することとした．例えば，図 4 に示すようにサイト 0 は最初から 3 クール計算可能であるが，4 クール目は計算できないようになる．そして停止後，再び計算を行うことができる場合もある．この時，前回までの探索結果を残しているかどうかは重要になる．以降で行うシミュレーション実験では，前回までの探索結果を使う場合はプログラムに探索データの保存機能が付いている，あるいは停止したクール数だけその計算拠点から計算結果が返ってくるのが遅れたと想定している．計算結果が遅れて返ってくる理由としては，計算資源の能力差やネットワークのトラフィックの問題などが考えられる．

各クールの最後には，計算を行った各計算拠点が全ノードのエリート個体を集めて，それを管理プロセスを実行する計算拠点に送信するという処

理を行うことにした。集められたエリートは、いったん管理プロセスのキューに保存される。そして次のルールに計算を行う計算拠点は、ルールの最初にこの管理プロセスのキューからエリート個体をノード数分受け取ることにする。ただし、各サイトの1つ以上のノードはそのチェックポイント時において最も適合度値の高い個体を受け取ることにした。

4.2 数値実験

前節で提案したモデルについての数値実験をシミュレーションとして行った。想定するグローバルコンピューティング環境は、図5に示す通りである。すなわち複数の計算拠点が存在し、各計算拠点はクラスタとなっている。対象問題には、前章と同じく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。

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

Dual DGAを行う時の各種パラメータは、表1のものを用いる。また、チェックポイントは100世代毎に設ける。

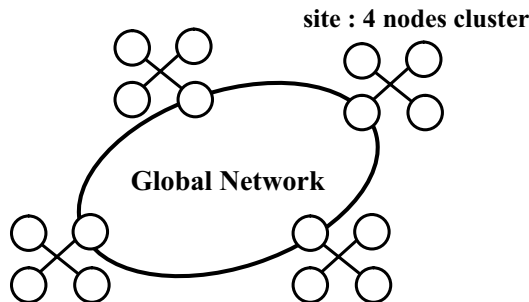


図5: 4 sites (4 nodes / site)

4.3 キューの効果

提案モデルのキューの効果について検討を行った。1つのサイトあたりのノード数を4として、4サイト(計16ノード)を使って計算を行った。キューの仕組みを使った場合の実験結果を図6に、キューの仕組みを使わなかった場合の実験結果を図7に示す。これら2つの実験結果を比較すると、キューの仕組みを使った方がより高速に解を発見できているのが分かる。

表1: Parameter setting of the Dual DGA

島数	192
プロセッサ数	4
コーディング	10bit グレイコード
交叉	1点交叉
突然変異率	1 / 遺伝子長
移住トポロジ (内, 外)	リング
移住間隔 (内)	5
移住間隔 (外)	Intra migration gap * 5
移住率 (外)	0.1 (最低1島)

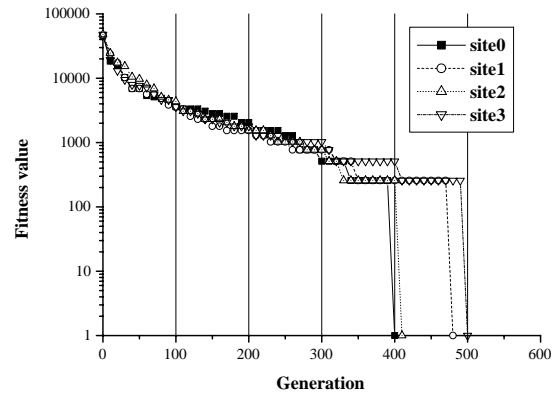


図6: Using a queue

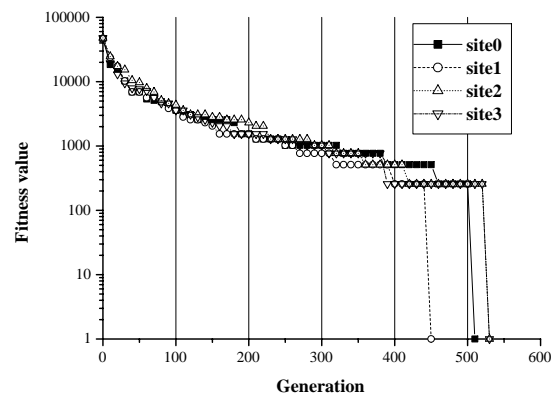


図7: Not using a queue

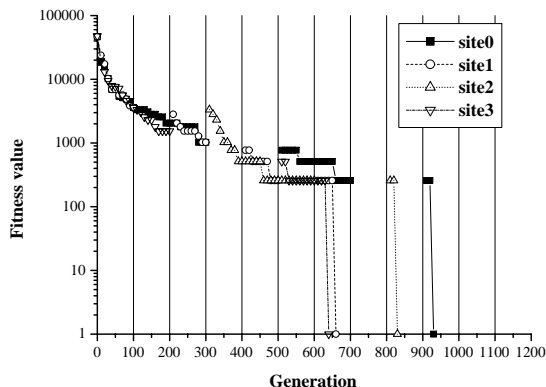


図 8: Delay (using a queue)

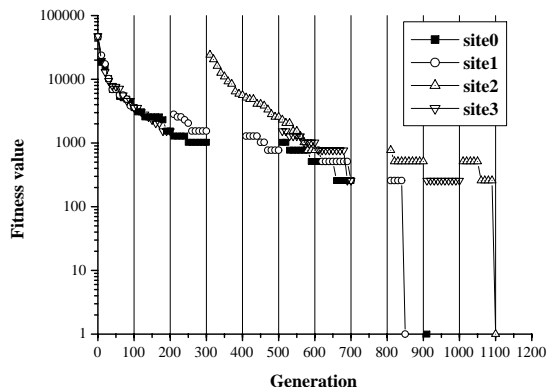


図 9: Delay (not using a queue)

4.4 遅延・再開

2章で説明したようなグローバルコンピューティング環境の特徴から，計算の途中で次のような局面に遭遇することが考えられる．遠隔地に存在する各サイトの計算能力は非均質であったり，ネットワーク的距離もそれぞれ大きく異なる．これにより，計算結果が返ってくるまでに何らかの遅延が発生してしまう．また，突発的に起きるサイトの障害やサイトにつながるまでの途中のネットワークの障害などにより，あるサイトの計算を緊急に停止させなければならない状況が生じる可能性もある．以下の実験は，これらの状況に対応できるかどうかを検討するために行った．

本シミュレーションでは，計算結果が返ってくるまでの遅延はクール単位で指定する．すなわち，対象となるサイトのあるクール数の間停止させ，停止させている間も個体を格納しているメモリは開放しない．一方，サイトが緊急停止する場合をシミュレートする場合には，復旧後に再び計算を行うと前に個体を格納しているメモリを初期化する．結果として，探索を初めからやり直すことになる．

図 8，図 9 は遅延をシミュレートした実験結果である．図 8 はキューの仕組みを使った場合，図 9 はキューの仕組みを使わなかった場合である．図中のグラフの直線が途切れている部分が，各サイトで遅延が発生している箇所である．

図 10，図 11 は復旧後のやり直しをシミュレートした実験結果である．図 10 はキューの仕組みを使った場合，図 11 はキューの仕組みを使わなかった場合である．図中のグラフの直線が途切れている部分が，各サイトが緊急停止されている箇所である．

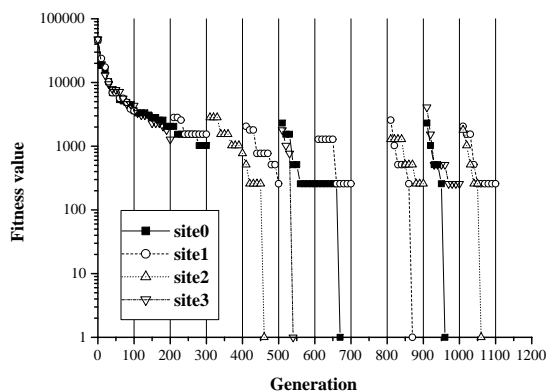


図 10: Restart (using a queue)

ある．

これらの結果より，いずれの場合にもキューの仕組みが有効に働いているのが分かる．すなわち，キューから他のサイトのエリート個体を受け取ることで，探索が遅れているサイトはその遅れを取り戻すことができるのである．

4.5 計算資源の追加

図 12 は 3 種類の実験結果を示している．いずれの結果も 2 クール目までは site0 が単独で探索を行う．single は site0 がそれ以降も単独で探索を続けた結果である．succeed は 2 クール目が終わった時点で site0 が停止し，site1 が site0 の探索を引き継いで，単独で探索を継続した結果である．この場合 site1 はキューの仕組みにより，site0 の過去のエリート個体を受け取ることができる．これにより site1 は，数十世代後に site0 が探索を継続できた場合とほぼ同じスピードで探索を行うことができ

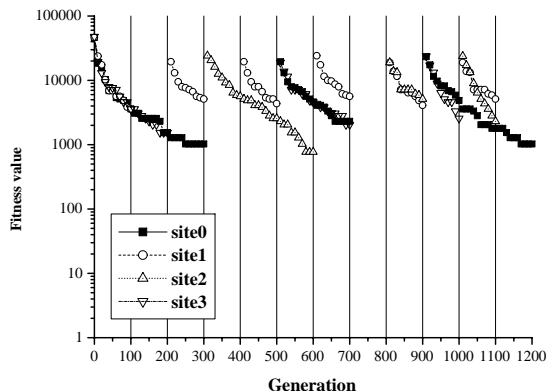


図 11: Restart (not using a queue)

るようになっている。また、site1 が探索を継続した直後に適合度値が上昇しているのは、キューから得たエリートがそのまま site1 のエリートとはならず、いずれかの個体と交叉を行うからである。

そして add は、2 クール目まで site0 が単独で探索を行い、それ以降はこれに新たに 2 サイトが加わって計 3 サイトで探索を行った時の最も早く解を見つけたサイトの探索の様子を示している。計算資源を増やしたこととキューの仕組みにより、解の探索速度が早くなっているのが分かる。

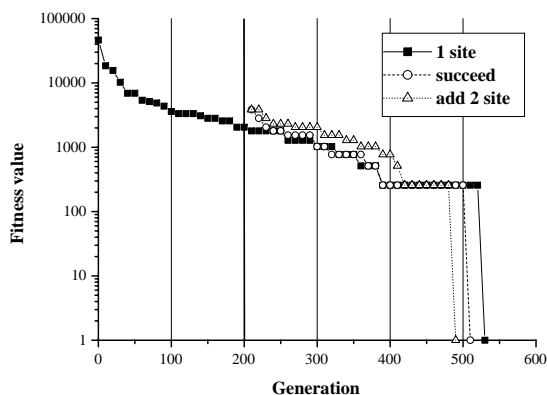


図 12: Succeed, Add site

この結果より、提案するモデルでは、1つのサイトが緊急停止した場合にでも探索を継続できること、途中で追加した計算資源を有効に利用できていることが可能となっている。

5 結論

本論文では、遠隔地に存在する複数のクラスタを用いた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 Dual DGA のモデルについて検討を行った。我々が提案したモデルでは、チェックポイントとキューの仕組みを用いることにより、計算資源の構成が動的に変化した場合への対応がスムーズに行えること、より多くの計算資源を用いることでより短時間に良好な解を求められることを確認した。具体的には、以下に示す事項である。

- i. 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。
- ii. サイトの緊急停止や遅延に対してロバスト性がある。
- iii. 追加した計算資源を有効に利用できる。

参考文献

- 1) 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
- 2) Enrique Alba, Jose M. Troya, "A Survey of Parallel Distributed Genetic Algorithms", Complexity Vol.4 No.4, John Wiley & Sons, Inc., 1999
- 3) D.E. Goldberg, "Genetic algorithms in search, optimization and machine learning, Addison-Wesley, 1989
- 4) Ian Foster, Carl Kesselman, "The Grid: Blueprint for a New Computing Infrastructure", Morgan Kaufmann, 1998
- 5) Andrew S. Grimshaw, William A. Wulf, James C. French, Alfred C. Weaver, Paul F. Reynolds Jr., "A Synopsis of the Legion Project", UVA CS Technical Report CS-94-20, 1994
- 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) 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
- 8) Rajkumar Buyya, "High Performance Cluster Computing, Vol.2, Programming and Applications", Prentice Hall, 1999
- 9) Kai Hwang, "Scalable Parallel Computing", WCB/McGraw-Hill, 1998