

Grid 計算環境における遺伝的アルゴリズムの提案

谷村 勇輔[†] 廣安 知之^{††} 三木 光範^{††}

本研究では Grid 計算環境における遺伝的アルゴリズム (GA) の提案を行った。提案モデルは GA の特性を考慮した仕組みを備えたマスタ・ワーカー型である。マスタサイトは、ワーカー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、マスタサイトの持つキューに確保する。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。このようなメカニズムを有することで Grid 計算環境で生じるであろう突然のサイトの停止やネットワークの切断にも柔軟に対応できると考えられる。本研究では提案モデルの有効性を検討するために、仮想的な Grid 環境を用意し、シミュレーション実験を行った。その結果、提案モデルがそうした Grid 環境の障害に対してロバストであることが確認できた。次に 2 サイトを使用し、構築したシステムの通信部分について検討を行った。

Genetic Algorithms on the Computational Grids

YUSUKE TANIMURA,[†] TOMOYUKI HIROYASU^{††} and MITSUNORI MIKI^{††}

In this study, Genetic Algorithms (GA) on the computational grids is discussed. The proposed system is a master worker model with a mechanism which takes the feature of GA into consideration. 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 among the worker sites. Because of this mechanism, the proposed system can solve the problems on the computational grid; the sudden halt of the site and network. In this study, we prepared a virtual Grid environment and evaluated the proposed system on it as a simulation. Then, we practically developed the proposed system and confirmed if the system works normally or not.

1. はじめに

コンピュータの性能は飛躍的に向上したが、科学技術計算に代表される大規模計算を行なうにはまだ不十分である。また、1つの機関が手に入れることのできるコンピュータ・パワーも限られている。そこで、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。特に、次世代の社会的基盤となるべき強力なグローバル・コンピュータ・ネットワークは Grid と呼ばれ、その構築方法や利用方法に関する研究が欧米を中心に進められている⁶⁾¹⁹⁾²⁰⁾。Grid に関する研究は、現在テストベッドを構築・維持するという形

で推し進められている。そして Globus⁷⁾ や Legion¹⁰⁾ といった Grid 環境を構築するための基本システムが開発されている。このように Grid を実現する環境は整いつつある。しかし一方では、Grid 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⁴⁾ や佐藤らの netCFD の研究¹⁸⁾¹⁷⁾ などがあるが、ほとんど行われていない。

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。最適化は構造物設計¹⁶⁾、タンパク質の構造解析¹⁴⁾ などの際に利用される技術で、解析を繰り返す行うために非常に計算コストが高くなる。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。その1つの解決方法が最適化の並列処理であり、Grid 環境

[†] 同志社大学大学院工学研究科
Graduate School of Engineering, Doshisha University
^{††} 同志社大学工学部
Department of Engineering, Doshisha University

下で最適化を行なうことができれば大規模な問題の解決につながる。2章で説明するが、Grid 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。つまり Grid 環境を利用して最適化計算を行うためには、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 (Genetic Algorithms, 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⁸⁾。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存在すると考えられる。このような特徴は、Grid 環境を利用して最適化計算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、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。検討する GA モデルは、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)³⁾ を、Grid 環境に拡張したモデルである。Dual DGA は分散 GA の拡張モデルであり、解探索能力に優れ、ユーザが設定すべきパラメータの数も分散 GA と比較して少ない。さらに、本研究では検討していないが、島数が多いために島数によりロードバランスをとることも容易である。

構築したシステムはマスタ・ワーカ型である⁹⁾⁵⁾²¹⁾。本研究では Grid 環境としてハイパークラスタを想定し、各クラスタ型並列計算機¹⁾²⁾ をサイトと呼ぶ。これらのサイトのうち、ある 1 つのサイトをマスタサイトとし、残りをワーカサイトとする。マスタサイトは各ワーカサイトに対して Dual DGA のパラメータを与え Dual DGA を起動する。ワーカサイトでは Dual DGA が実行され、探索が行なわれる。マスタサイトは一定時間間隔でワーカサイトの状態をチェックし、その際にワーカサイト毎の最良探索点の状態を取得する。取得された探索点は、マスタサイトにより他ワーカサイトに対して任意に分配される。この操作により最適解は高速に求まる。同時に、あるワーカサイトが停止し、それまでの計算が強制的に終了されてしまっても、マスタサイトが取得している他ワーカサイトの探索情報を利用して再開することで、効率的な計算が期待できる。これは Grid 環境において予想されるサイトやネットワークの障害に対してロバストなモデルであるといえる。

提案モデルおよびシステムの検討は、擬似的に Grid 環境を構築し、いくつかのシナリオを想定した実験を

通して行なった。主な検討項目は 2 つあり、1 つは複数の計算資源を利用して高速に最適解を見つけられるかどうかの検討である。もう 1 つは、計算途中でサイトが停止したり、サイト間のネットワークが途切れたりするといった障害に対してロバストであるかどうかの検討である。

2. Grid 計算環境

Grid 計算環境は、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¹¹⁾。

- (1) 計算は常にネットワークを介して行なわれる。
- (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なわれる。

これは Grid 計算環境を利用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場合がある (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場合がある (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、Grid 計算環境は多くの場合、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⁸⁾。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。

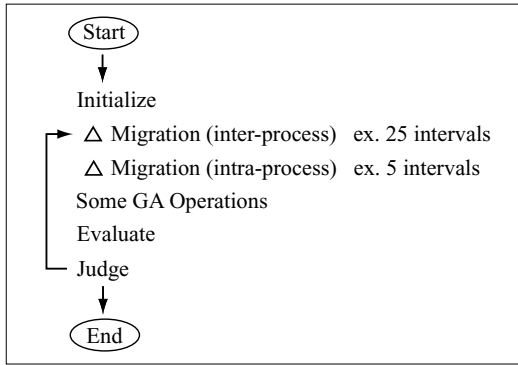


図1 並列 Dual DGA の計算の流れ
Fig.1 Flow of Parallel Dual DGA

GA の並列モデルの 1 つに島モデル (分散 GA) がある。島モデルは母集団を複数の島に分けてそれぞれ GA を適用し、ある世代間隔で移住と呼ばれる解交換を行いながら探索を行うモデルである。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^{23) 15) 12)}。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¹³⁾。Dual DGA では、島内の個体数を 2 と設定する。これによりパラメータの 1 つである交叉率を 1.0、移住率を 0.5 と一意に決定でき、アルゴリズムを簡易化できる。さらに Dual DGA では、できるだけ全体の多様性が維持できるように、交叉や突然変異を独自の方法で行なう。

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²⁴⁾。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 のプロセスを 1 つ起動する。そして 2 段階の移住を行う。同一プロセス内の移住では島間で個体の交換を行うが、異なるプロセス間の移住では島の交換を行う。図 1 に計算の流れを示す。このような階層的な移住モデルを採用することにより、実際にプロセス間通信を必要とする上位の移住を減らすことができる。本研究では、プロセス内の移住間隔を 5 世代、プロセス間の移住間隔を 25 世代と設定している。

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 関数の最小化問題を解いた結果を示す。実験で用いた Ridge 関数の次元数は $30(n = 30)$ である。

表 1 Dual DGA のパラメータ設定
Table 1 Parameter setting for Dual DGA

Number of islands	192
Number of processes (in parallel)	4
Coding method	10bit, graycode
Crossover method	1pt crossover
Mutation Rate	1/Length of the gene
Migration topology (intra, inter)	Ring
Intramigration gap	5
Intermigration gap	Intramigration gap × 5
Intermigration rate	0.1 (1 island at least)

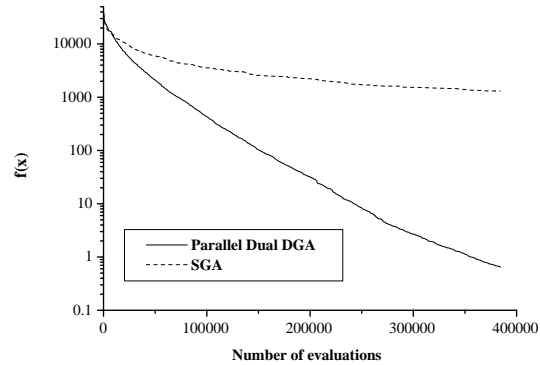


図2 SGA と並列 Dual DGA の比較
Fig.2 Comparison of SGA with Parallel Dual DGA

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

用いたパラメータは表 1 に示す通りである。図 2 に並列 Dual DGA と単純 GA (SGA) 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。本論文では、評価回数は (世代数 × 個体数) で表され、およそその計算量を示すことになる。並列 Dual DGA の評価回数は全プロセッサの評価回数の合計である。すなわち、図 2 では、並列 Dual DGA は SGA より少ない計算量 (評価回数) で解が求まっており、効率的に探索を行っているといえる。この傾向は、GA が解探索困難な問題以外のいくつかのテスト関数に対して同様であった。

このように Dual DGA の利点は解探索能力が高いことである。また、並列計算機に実装した際の通信負荷を少なくできる可能性がある。第 1 の理由は、先に述べた階層的な移住モデルの採用である。第 2 の理由は並列化の粒度を柔軟に変更できる点である。並列 Dual DGA では島数とプロセッサ数を多対 1 で対応させるが、これは実行する並列計算機の (通信性能 / 計算性能) の割合によって変更可能である。こうした

利点は、Grid 環境のように異なるスペックをもつ多数のクラスタを利用する際に重要となると考える。もちろん、各プロセッサに割り当てる島数やプロセス間移住の間隔を変更することで、解探索に影響がないわけではない。しかしながら階層的な移住モデルを用いることにより、そうした影響は小さくなると考えている¹³⁾。

4. Grid 計算環境における Dual DGA のモデル

本研究では、Grid 計算環境を利用できる GA の検討を行う。検討モデルは Dual DGA をベースに、複数の計算拠点をを用いて並列に探索を行うモデルである。計算拠点は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とにする。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。これをマスタサイトと呼ぶ。マスタサイト以外のサイトはワーカサイトと呼び、それぞれ並列 Dual DGA を実行する。つまり、検討モデルは図 3 に示すように計算が行われることになり、マスタ・ワーカ型のモデルであるといえる。

マスタサイトで実行する管理プロセスは以下の機能を有する。ただし(1)や(2)は Grid のジョブマネージャ等が実装すべき機能であり³⁾²²⁾、管理プロセスはそれらと協調して動作することが望まれる。そこで本論文ではこれらの詳細については考えない。これらを考慮したモデルの検討は今後の課題である。

- (1) ワーカ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) ワーカサイトに対して Dual DGA の実行ジョブを投入する。

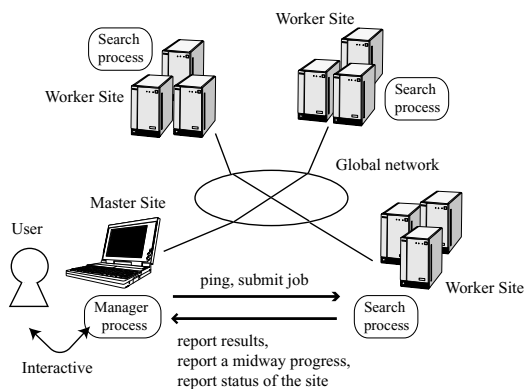


図 3 検討モデルの概要

Fig. 3 Concept of the master-worker based GA on the computational grids

- (3) ワーカサイトで実行された探索プロセスと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。

検討モデルは次のように計算を行う。管理プロセスは、まず最初にワーカ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、管理プロセスはワーカサイトに関する情報を取得する(1)。計算を続ける場合には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 での個体)を取得する(3)。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(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無制限としている。

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。そして、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

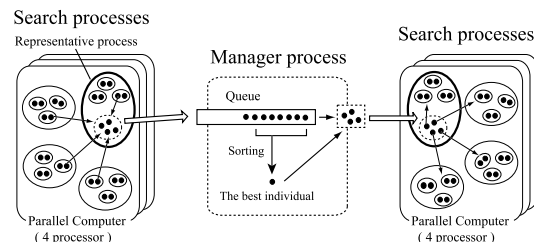


図 4 キュー操作の仕組み

Fig. 4 Operation in the master's queue

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4章で説明した検討モデルについて評価実験を行なった。本章では、シミュレーションによる実験と実際のシステムを構築して行なった実験の結果を示す。しかしながら、いずれの実験も実際の Grid 環境下で行なわず、実際の Grid 環境を想定した3つのシナリオをもとに行なった。

(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。本シナリオは、何も障害が起きない場合に検討モデルが有効であるかどうかを確認するために用意した。

(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、チェックポイントを定期的にとることのできないサイトが計算に参加することの有効性を確認するために用意した。本シナリオはタイムステップ毎に乱数を発生させて、ネットワークが使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。本シナリオは計算を完了するまでに何日もかかるような場合を想定し、検討モデルが障害に対してロバストであるかどうかを確認するために用意した。

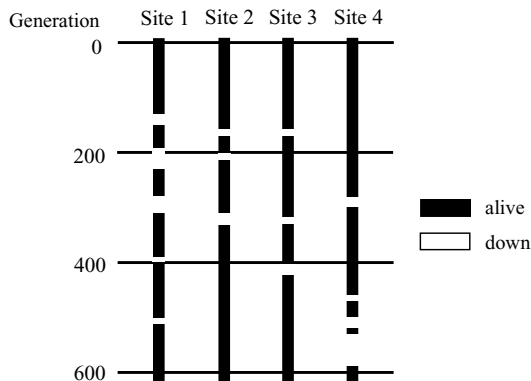


図5 シナリオ3の例
Fig.5 Example of Scenario 3

例えば、図5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と同様にタイムステップ毎に乱数を発生させて、その時間帯にサイトが稼働しているかどうかを確率的に決定することで作成した。また、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。

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。ただし、1つのワーカサイトが実行するGAの総島数は96島とした。

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

5.3 シミュレーション実験の結果

提案しているDual DGAのシステムが想定通りの挙動を行なうかどうかの検討を次に示すシミュレーション環境で行なった。本実験では、1つのマスタサイトと等しい計算能力をもつ4つのワーカサイトが利用できるGrid環境を想定した。各ワーカサイトは4プロセッサの並列計算機からなる。実際のシミュレーションでは16プロセッサのクラスタ型並列計算機を用いた。これを仮想的に4分割し、シナリオに従ってプログラムの通信をやめさせたり、計算をやめさせたりした。またマスタサイトの機能は、いずれかのプロセッサが受けもつことにした。以下に、シミュレーション実験の結果を述べる。

(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した。本実験では、チェックポイント時にキュー操作を行うモデルとキュー操作を行わないモデルについて比較した。用いたシナリオはシナリオ1とシナリオ2である。いずれのシナリオもチェックポイント間隔を100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30試行の平均を示す。表中のDRは、実際にチェックポイント時にネットワークが途切れてワーカサイトと接続できなかった割合である。これらの結果より、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見つけていることができていいるのが分かる。

(2) ワーカサイトの停止による影響

シナリオ3において、キュー操作を行うモデルと行な

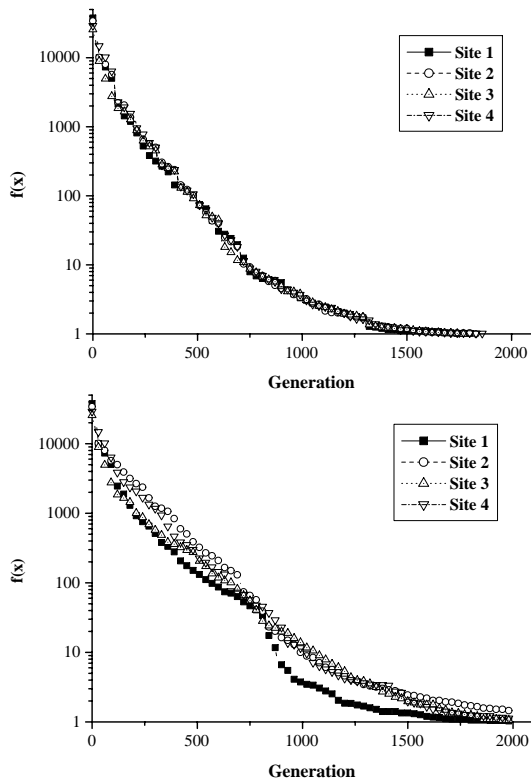


図 6 シナリオ 1 の結果例 (キュー操作あり (上), キュー操作なし (下))

Fig. 6 Results of the scenario 1 – an example

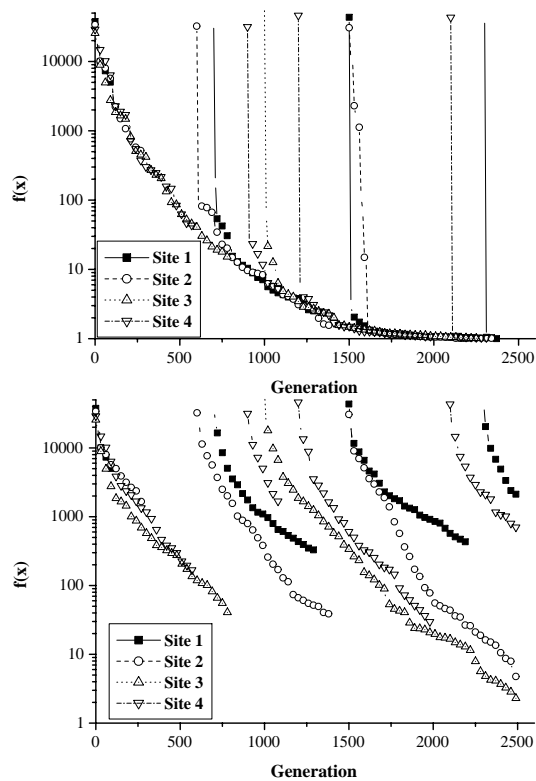


図 7 シナリオ 3 の結果例 (キュー操作あり (上), キュー操作なし (下))

Fig. 7 Result of the scenario 3 – an example

表 2 シナリオ 1, 2 の結果

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

表 3 シナリオ 3 の結果

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

わないモデルについて比較した。シナリオ 3 において、停止したサイトではそれまでの探索点情報が全て失われる。その後、サイトが復旧した場合はランダムに探索点を生成し、計算を再開する。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、30 試行の結果を表 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。

(3) 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、検討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 (1)(2) の実験結果が示しているように、検討モデルは高速に解探索を行うことができる。さらに、シナリオ 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、チェックポイント時のキュー操作は、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う役割をもつ。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ワーカサイトや、新たに計算に加わるワーカサイトが、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。

5.4 実際のシステムによる実験結果

Globus ツールキットを利用して、4章のモデルを実装した。システムアーキテクチャは図8に示す通り、Manager, Monitor, Searcher のコンポーネントからなる。計算は次のように行なう。ユーザはマスタサイトにおいて Manager プロセスを起動する。Manager は GRAM (Globus Resource Allocation Manager) を利用して、指定されたワーカサイトにおいて Monitor 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、Searcher の書き出したファイルを読み込み、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。この時、ワーカサイトにおいて GASS サーバが必要となるが、これは Monitor の役割である。Monitor はこれ以外にも、サイトの負荷を監視する役割を担う予定であるが、現在未実装である。

本システムを2つの並列計算機に導入し、実験を行なった。ワーカサイトはクラスタ型並列計算機であり、サイト1は16個の PentiumII (400MHz) からなり、サイト2は33個の PentiumIII (800MHz) からなる。

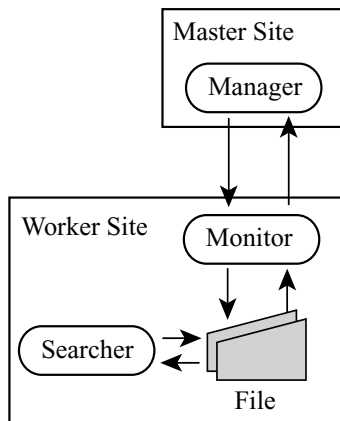


図8 システムアーキテクチャ
Fig. 8 System architecture

クラスタ内のネットワークは 100Mbps のイーサネットである。またサイト1はマスタサイトとワーカサイトを兼ねることができ、その際には、サイト1のある1つのプロセッサが Manager を実行する。提案する Dual DGA モデルの解探索性能の検討は前節で行なっているため、本節では構築したシステムが検討モデルをスムーズに実行できる性能を備えているか検討を行なう。

(1) マスタ・ワーカ間の通信時間

検討モデルを実行する際にマスタサイトとワーカサイトで行なわれる通信に要する時間を測定した。通信には、ワーカサイトからマスタサイトへの探索点情報の送信 (Read), マスタサイトからワーカサイトへの探索点情報の送信 (Write) の2種類がある。実験ではマスタサイトをサイト1、ワーカサイトをサイト2が担当する。サイト間のネットワークには 10BASE-TX, 100BASE-TX のイーサネットの両方を用意し、それぞれにおいて計測を行なった。なお、10BASE-TX を用いた時のネットワークのスループットは 6.9Mbps, 100BASE-TX を用いた時は 68.7Mbps であった。結果を図9に示す。図9においては横軸を探索プロセスとしている。検討モデルでは、ワーカサイト毎に実行する探索プロセスの数が増加すると、マスタサイトとやり取りする通信量が増加する。しかしながら図9の結果から分かるように、探索プロセスを増やしても通信時間にほとんど差はない。また 10BASE と 100BASE のイーサネットによる差も見られない。これは検討モデルが、非常に通信コストを抑えたモデルであるからである。

(2) チェックポイントに要する時間

各ワーカサイトで実行する探索プロセス数を増加させた時のチェックポイントに要する時間を調べた。この

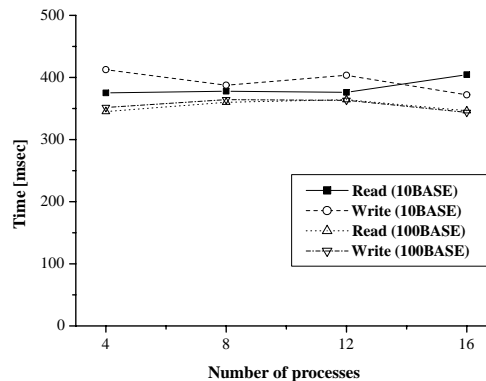


図9 通信に要する時間
Fig. 9 Cost of communication between a master and a worker

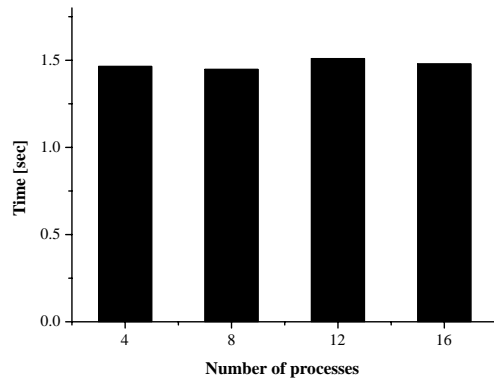


図 10 チェックポイントに要する時間
Fig. 10 Cost of the checkpoint

実験では、ワーカサイトはサイト 1 とサイト 2 の 2 つであり、両サイトの探索プロセスを同時に増加させる。結果を図 10 に示す。これによると、探索プロセス数にかかわらず、1 回のチェックポイントあたり約 1.5 秒の時間を要するのが分かる。これは構築したシステムがある程度のスケーラビリティをもつことを示している。対象とする最適化問題の目的関数を計算するのに要する時間が、1.5 秒より十分に大きい場合は検討モデルは有効に働くであろう。しかしながら、ワーカサイト数や探索プロセス数があまりにも大きくなり、チェックポイントに要する時間が目的関数を計算するのに要する時間を大きく上回るようになれば、各ワーカサイトで実行する GA の探索レベル(いわゆる進化の程度)が異なってくる。そのような場合に、検討モデルが有効であるかどうかは未知数であり、今後の課題といえる。

6. 結 論

本論文では、遠隔地に存在する複数の強力な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 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。我々が検討したモデルはマスタ・ワーカ型のモデルであり、アプリケーションレベルで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 Grid 環境におけるシミュレーションを通じて、検討モデルが以下の特徴を有することを確認した。

- (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) ワーカサイトが停止したり、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。

Grid 計算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、検討モデルは Grid 計算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。そして検討モデルを実際のシステムとして構築し、ごく基本的な実験を行なった。これにより小規模な資源が利用できる場合、検討モデルはサイト間の通信量が少なく、効率的に機能することが確認できた。今後は、より大規模な Grid 環境を構築し、スケーラビリティやシミュレーション実験の結果について検討していきたいと考えている。

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) Rajkumar Buyya. *High Performance Cluster Computing: Architecture and Systems*, Vol. 1. Prentice Hall, 1999.
- 2) Rajkumar Buyya. *High Performance Cluster Computing: Programming and Applications*, Vol. 2. Prentice Hall, 1999.
- 3) H. Casanova, G. Obertelli, F. Berman, and R. Wolski. The AppLeS Parameter Sweep Template: User-Level Middleware for the Grid. In *Proceedings of SC2000*, 2000.
- 4) 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.
- 5) J. Czyzyk, M. Mesnier, and J. Moré. NEOS: The Network-Enabled Optimization System. Technical Report MCS-P615-1096, Mathematics and Computer Science Division, Argonne National Laboratory, 1996.
- 6) Ian Foster and Carl Kesselman. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, 1998.
- 7) Ian Foster and Carl Kesselman. Globus: A Metacomputing Infrastructure Toolkit. *The International Journal of Supercomputer Applications and High Performance Computing*, Vol. 11, No. 2, pp. 115–128, 1997.
- 8) D. E. Goldberg. *Genetic algorithms in search, optimization and machine learning*, 1989.
- 9) J. Goux, S. Kulkarni, J. Linderoth, and M. Yoder. An Enabling Framework for Master-

- Worker Applications on the Computational Grid. In *Proceedings of HPDC-9*, pp. 43–50, 2000.
- 10) A. S. Grimshaw, W.A. Wulf, and the Legionteam. The Legion vision of a worldwide virtual computer. *Communications of the ACM*, Vol. 40, No. 1, pp. 39–45, 1997.
 - 11) 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*, Vol. 17, No. 99, 2000.
 - 12) Tomoyuki Hiroyasu, Mitsunori Miki, and Yusuke Tanimura. The Difference of Parallel Efficiency between the Two Models of Parallel Genetic Algorithms on PC Cluster Systems. In *Proceedings of the 4th HPC Asia*, Vol. 2, pp. 945–948, 2000.
 - 13) Tomoyuki Hiroyasu, Mitsunori 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.
 - 14) 木寺詔紀. コンピュータ解析 –最適化をめぐる。蛋白質の時代 –構造・物性・機能研究の新局面, pp. 1380–1386, 1994.
 - 15) B. Kröger, P. Schwenderling, and O. Vornberger. Parallel Genetic Packing of Rectangles. In *Proceedings of 1st PPSN*, 1990.
 - 16) Zbigniew Michlewicz. *Genetic Algorithms + Data Structure = Evolution Programs*. Springer-Verlag, 1992.
 - 17) Hidemoto Nakada, Mitsuhsa Sato, and Satoshi Sekiguchi. Design and Implementations of Ninf: toward a global computing infrastructure. *Future Generation Computing Systems, Metacomputing Issue*, Vol. 15, pp. 649–658, 1999.
 - 18) Mitsuhsa Sato, Kazuhiro Kusano, Hidemoto Nakada, Satoshi Sekiguchi, and Satoshi Matsuoka. netCFD: a Ninf CFD component for Global Computing, and its Java applet GUI. In *Proceedings of the 4th HPC Asia*, Vol. 1, pp. 501–506, 2000.
 - 19) 関口智嗣. グローバルコンピューティングとは. *Computer Today*, Vol. 17, No. 95, 2000.
 - 20) 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*, Vol. 17, No. 96, 2000.
 - 21) 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 -. 並列処理シンポジウム, pp. 277–284, 2000.
 - 22) 竹房あつ子, 合田憲人, 小川宏高, 中田秀基, 松岡聡, 佐藤三久, 関口智嗣, 長嶋雲兵. グローバルコンピューティングのスケジューリングのための性能評価システム. *情報処理学会論文誌*, Vol. 41, No. 5, pp. 1628–1638, 2000.
 - 23) Reiko Tanese. Parallel Genetic Algorithms for A Hypercube. In *Proceedings of the 2nd ICGA*, pp. 177–183, 1987.
 - 24) 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.

(平成 13 年 8 月 31 日受付)

(平成 14 年 2 月 13 日採録)

谷村 勇輔 (学生会員)

1976 年生 . 2001 年同志社大学大学院工学研究科修士課程修了 . 同年同志社大学大学院工学研究科博士課程入学 . クラスタや広域環境における並列・分散計算に興味を持つ .

IEEE-CS , 超並列計算研究会各学生会員 .

廣安 知之 (正会員)

1966 年生 . 1997 年早稲田大学理工学研究科後期博士課程終了 . 2001 年より同志社大学工学部専任講師 . 進化的計算 , 最適設計 , 並列処理 , 設計工学などの研究に従事 . IEEE , 電気情報通信学会 , 計測自動制御学会 , 日本機械学会 , 超並列計算研究会 , 日本計算工学会各会員 .

三木 光範 (正会員)

1950 年生 . 1978 年大阪市立大学大学院工学研究科博士課程終了 , 工学博士 . 大阪市立工業研究所研究員 , 金沢工業大学助教授を経て 1987 年大阪府立大学工学部航空宇宙工学科助教教授 , 1994 年同志社大学工学部教授 . 進化的計算手法とその並列化 , および知的なシステム的设计に関する研究に従事 . 著書は「工学問題を解決する適応化・知能化・最適化法」(技法堂出版)等多数 . IEEE , 米国航空宇宙学会 , 人工知能学会 , システム制御情報学会 , 日本機械学会 , 計算工学会 , 日本航空宇宙学会各会員 . 超並列計算研究会代表 .