統計数理(2004)
第 52 巻 第 2 号 381–391

［研究ノート］

グリッド環境に適した遺伝的アルゴリズムによる最適化

染谷 博司†

(受付 2004 年 2 月 10 日；改訂 2004 年 6 月 14 日)

要　　旨

近年，最適化手法としての遺伝的アルゴリズム(GA)の有効性が報告されている．しかし，GA は多大な計算量を必要とする．本稿では，GA の計算資源としての計算グリッド環境に着目し，計算グリッド環境への GA の適用について述べる．まず，GA が考慮すべきグリッドの特徴を述べ，グリッド環境に適した GA およびその設計について考察した．また，遠隔地間を接続するグリッド環境での GA の実装例および最適化問題への応用例を示し，その有効性を確認した．

キーワード：遺伝的アルゴリズム，グリッド，分散計算，並列計算，最適化．

1. はじめに

困難な最適化問題に対する有望な最適化手法としての遺伝的アルゴリズム(Genetic Algorithm; GA)の有効性が報告されている．GA は，明示的な勾配情報を必要としない直接探索法であるため適切な評価関数の設定ができれば，非線形関数や不連続な関数など様々な最適化問題に適用可能である．また，シミュレーテッド・アニーリング(Simulated Annealing; SA)などの他の確率的最適化手法と比較した場合，集団探索や交叉といった特徴を持ち，多峰性が強く最適化が困難とされている問題に対しても高い最適化性能を示す．しかしその一方，GA は試行錯誤の繰り返しにより探索を行うため一般に多大な計算量を必要とする．

本稿では，GA の計算資源として計算グリッド環境に着目する．まず，GA が考慮すべきグリッドの特徴を述べ，グリッド環境に適した GA およびその設計について考察する．次に，グリッド環境に適した GA のひとつであると考えられる *Genetic algorithm with Search area Adaptation* (GSA)について概括する．また，遠隔地間を接続するグリッド環境での GA の実装例および最適化問題への応用例を示し，その有効性について議論する．

2. グリッド環境に適した遺伝的アルゴリズム

2.1 グリッドの特徴

インターネットなどの広域ネットワーク環境上の膨大な各種資源を共有利用するグリッド技術が注目されている (Foster et al. (2001))．特に，計算資源の共有利用を目的とした計算グリッド (Computational Grid)では，多くの時間がアイドル状態にあるとされるプロセッサを共有し有効活用することにより，膨大な計算資源を利用可能にする．本稿では，以後，単にグ

† 統計数理研究所：〒106–8569 東京都港区南麻布 4–6–7

リッドと言った場合には計算グリッドを指すものとする．

並列計算を行う既存の計算機システムには PC クラスタや並列計算機などがある．これらと比較し，グリッドには，計算時に考慮すべき特徴として以下のものが挙げられる（Baker et al.（2000），染谷（2003，2004））．

(1) セキュリティ … ユーザ認証，通信回線に対するセキュリティ，ファイアウォール越え，など．
(2) 不均質性 … サーバのアーキテクチャや OS，利用可能なコンパイラやインタプリタなどがそれぞれ異なる．
(3) 不確実性 … 利用可能なサーバや通信回線が動的に変化し，応答が返ってこないこともある．
(4) 非同期性 … 異性能の混合．サーバの処理速度や主記憶容量，通信回線の速度や帯域などがそれぞれ異なる．
(5) 超並列性 … 数百または数千程度以上の超並列性の実現が可能．

なお，本稿では，計算資源を提供しグリッドを構成する計算機等をサーバと呼ぶ．(1)～(3) については，ミドルウェアや OS レベルでの改良が進められている．セキュリティの改善には，ミドルウェアのひとつである Globus の GSI によるシングルサインオン認証や，通信路を暗号化した NFS が提案されている（The Globus Project，武田 他（2003））．不均質性については，Java Virtual Machine などの仮想マシン上でのバイトコードやインタプリタ言語を使用したり，必要な言語を有するサーバのみをミドルウェアが選択して使用するなどの工夫により吸収することができる．不確実性は，異なるサーバに冗長にジョブを依頼したり，一定時間応答のないジョブを別のサーバに再依頼するなどの方法により，非同期性に集約することができる．一方，非同期性および超並列性に関しては，アプリケーションレベルでの工夫が必要である．具体的には，サーバごとの計算処理を他の処理に対する局所性および独立性の高いジョブに割り当てたり，大規模な並列実行処理を実装可能とすることが必要である．

2.2 設計方針

グリッド環境において実装される GA は，非同期性および超並列性について考慮されていなければならない．各サーバに割り当てられる GA の処理単位には次のものが考えられる．

(1) 島モデル型 GA におけるサブ集団内の処理
(2) ファミリー内の処理
(3) 個体評価

(1)の方法は，島モデル型 GA に適用可能な方法である．島モデル型 GA は，集団全体をいくつかのサブ集団に分割し，それぞれを独立に進化させることで GA にとって重要とされる集団の多様性維持の効果を高めた手法である．実現可能な並列数は島の数以下であり，また，島の数は一般に数十程度以下であるため，グリッドの超並列性を活用するためには(2)および(3)の方法を併用する必要がある．島モデル型 GA では一般に，ある一定のタイミングでいくつかの個体を島間で交換する移住と呼ばれる操作を行う．非同期性が強い場合は，島間での進化速度の差による移住への影響を考慮する必要がある．

(2)の方法は，ファミリーごとに局所化された世代交代モデルを用いる GA に適した方法である．ここで「ファミリー」とは，一組の親個体およびこれらから生成される子個体等により構成される個体集団を指す．一般に，ファミリーは非同期な世代交代が可能であるため，集団サイズと同程度のオーダーの並列数が実現可能である．また，ファミリー内の個体生成や評価

の処理負荷が比較的大きい場合は，サーバとの通信頻度を相対的に少なくすることができる．集団サイズは一般に数百から数千以下であり，より大きな並列性が要求される場合には(3)の方法との併用が望ましい．

(3)の方法は，さまざまなGAモデルに対して幅広く適用可能である．しかし，個体評価の順序に依存関係がある場合は，同期のための待ち時間の影響により並列数ほどの計算時間短縮の効果が現れない場合もあり，非同期性を考慮した世代交代モデルとの組合せが必要である．なお，一個体の評価毎に通信が発生するため，評価のための計算時間と通信の遅延時間とのバランスを考慮した工夫が必要になる場合もある．

これらの他に，通信についても考慮する必要がある．グリッドでは，高速ネットワークを利用することを前提としているものの，並列計算機と比べてその通信の遅延時間は大きい．遅延の影響を軽減するためには，通信量が小さく通信頻度が少ないことが望ましい．また，当然のことながら，GAは高性能であることが望ましい．ここで性能とは，必要とする評価回数に対する解品質の良好さを指す．グリッド上のGAが非並列な環境で実装されるGAと比べ計算時間を短縮できるとしても，得られる解品質が劣るのでは優れたGAであるとは言えない．非並列な環境で実装されるGAと同程度の性能を維持したままで計算時間を短縮できることが望ましい．

以上により，グリッド環境に適したGAとは次のような特徴を有したGAであると考えられる．

- 世代交代が局所化されている
- 通信量が小さく通信頻度が少ない
- 高性能である

3. GSA

著者らは，適応的な探索をする進化型計算の一手法として，*Genetic algorithm with Search area Adaptation*（GSA）を提案している．GSAは，困難な組合せ最適化問題のひとつとして知られるフロアプラン設計問題（VLSIレイアウト設計問題）および関数最適化問題における代表的ないくつかのベンチマーク問題において，その有効性が確認されている（染谷・山村（1999, 2002），染谷（2001））．GSAは，グリッド環境にて実行されるGAが備えるべき特徴を有している．本章ではGSAについて概括する．

3.1 探索の方法

GSAでは，親個体集団の周囲の探索空間の景観を推測し，景観に対して適応的に探索領域を調整しながら探索を行う．探索の例を図1に示す．まず,(1)親個体集団の周囲に少数のサンプルを取り（図中(a)），それらを元に親個体集団の周囲の探索空間の景観を推測する（図中(b)）．ただし親個体の適応度は既知であるので，親個体間およびその外側からサンプルを取る．次に,(2)推測された景観に基づき，より密に探索すべきであると考えられる領域を集中的に探索する（図中(c)）．これにより，探索空間の景観を探索局面に応じて適切に仮定することができ，景観に応じた適応的で効率的な探索を行うことができるものと考えられる．

ここで，この探索方法を実現するための探索オペレータの機能分担について形質遺伝性の観点から考察を行う（山村 他（1992），小林（1993））．図2に，探索空間において交叉または突然変異によって生成される個体が，親個体または変異元の個体から，どのように形質を受け継いでいるかを示す．図中のArea Aは，親1からも親2からも離れた領域であり，また，親個体集団をひとつの分布とみなしたときにも，親個体分布からはずれた領域である．そのた

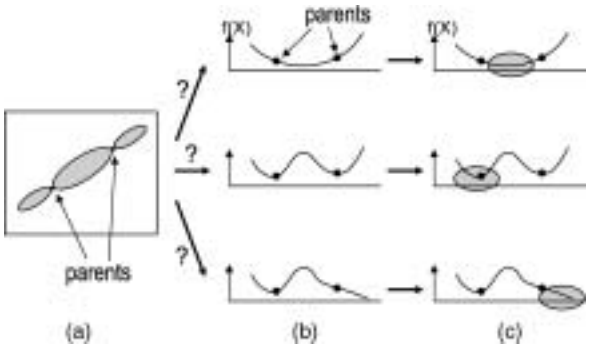

図 1. GSA **による探索.**

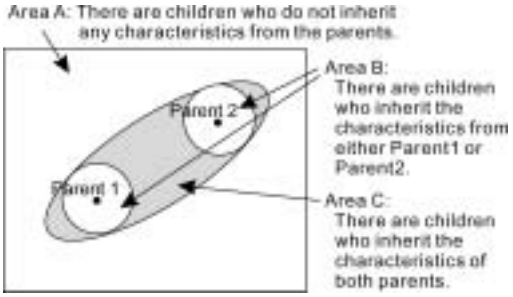

図 2. **形質遺伝性に基づいた探索領域の分類.**

め, Area A **に生成される個体は,両親のいずれからも形質を受け継いでいないと考えられる.** Area B **は,山登り法など多点探索を行わない確率的探索手法で用いられる探索オペレータが探索領域とする領域である.複数の個体情報を用いずともこの領域の探索が可能であることから,** Area B **に生成される個体は,親個体のどちらか一方の形質を主に受け継いでいると考えることができる.これに対し** Area C **は,複数の親個体の情報を用いることで初めて探索領域とされる領域である.そのため,親個体の両方からの形質を受け継いだ個体が生成される領域であると考えられる.以上の議論および交叉は複数の親個体を用いる探索であり突然変異は単独の個体を用いる探索であることを考慮すると,各探索オペレータの特徴を活かすためには,(**1**)の探索には交叉を,(**2**)の探索には突然変異を用いるべきであると考えられる.**

3.2 手順

GSA **は,「交叉による探索」と「突然変異による探索」から構成され,このループを** 1 **世代として繰り返す.** GSA **の世代交代モデルを図** 3 **に示す.**

交叉による探索の手順を以下に示す.

Step 1. **集団から一様分布に従いランダムに** 3 **個体を親個体として選択し,組合せを変えて** 3 **組のペアを作る.**

Step 2. **各ペアについて,次の** *Step* 3∼*Step* 6 **を行う.**

Step 3. $f_a = 50, f_b = 50$ **とする.**

Step 4. **交叉オペレータを用いて,両親からそれぞれ** f_a**%,** f_b**%の形質を受け継いだ子個体**

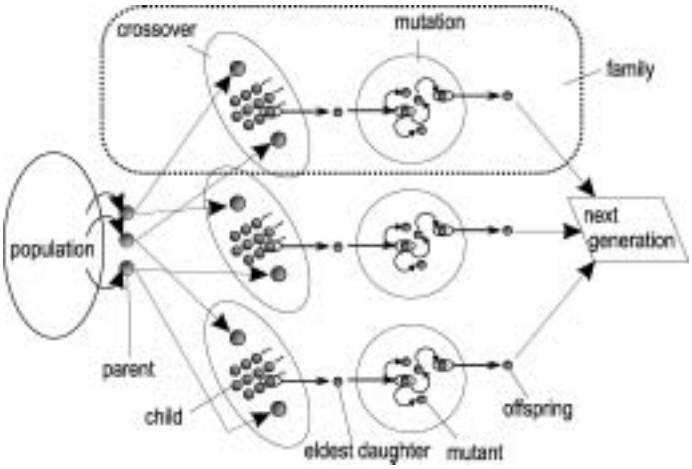

図 3. GSA の世代交代モデル.

を一定数生成する.

Step 5. 最良の子個体が最悪個体よりも不良な場合，f_a を決められた割合だけ減らし，また，f_b を決められた割合だけ増やし，*Step* 4 へ戻る.

Step 6. 最良の子個体を長女個体として選択する.

最悪個体とは，集団中で最も評価値の不良な個体である．長女個体とは，突然変異による探索の変異元となる個体である.

突然変異による探索の手順を以下に示す.

Step 1. 各長女個体に対して，*Step* 2∼*Step* 6 を行う.

Step 2. $i = 0$，$j = 0$ とする．長女個体を変異個体 $[0]$ とする.

Step 3. 突然変異オペレータを用いて，変異個体 $[i]$ から変異個体 $[i+1]$ を生成する．j に 1 を加える.

Step 4. 変異個体 $[i+1]$ の評価値が変異個体 $[0]$ の評価値よりも良好ならば i に 1 を加える.

Step 5. $j \leq R_m$（R_m はパラメータ）ならば *Step* 3 に戻る.

Step 6. すべての変異個体の中で最良の変異個体が最悪個体よりも良好ならば，それを次世代個体として選択する．不良であれば，集団内すべての個体および最良個体の「集団サイズ + 1」個の個体の中から，一様分布に従いランダムに選んだ個体の染色体を変異個体 $[0]$ にコピーし，それを次世代個体として選択する.

Step 7. 3 つの親個体と 3 つの次世代個体を入れ替える.

変異個体とは，突然変異によって生成される個体を指す．最良個体とは，探索開始からその探索時点までに得られた最も評価値の良好な個体を指す．次世代個体とは，変異個体中から選択され次世代に残される個体である.

交叉による探索では，両親の形質を均等に受け継いだ子個体を生成することで図 1 (a)に示す親個体周辺を大まかに探索し，より密に探索すべき領域を決定する．この決定は最悪個体を基準に行われるため，集団の探索レベルに応じた適応的な探索領域調整が行われる．なお，世代交代は局所化されているため，最悪個体は緩やかに交代する．突然変異による探索では，よ

り密に探索すべきであるとみなされた図 1 (c)に示す長女個体の周辺を集中的に探索する．ただし，探索領域の景観に適した効率のよい探索のために，長女個体よりも不良な方向への探索は行わない．その一方，変異個体 $[i]$ は長女個体よりも良好であれば変異個体 $[i-1]$ に対して改悪が許されており，小さな局所解に捕らわれない探索が可能である．突然変異による探索を行っても最悪個体よりも良好な個体が発見されない場合には，大きな局所解に陥っているとみなし，その領域に見切りをつける．

3.3 世代交代モデルの特徴

GSA の世代交代モデルには，形質の多様性維持に注目したいくつかの特徴がある．GSA では親子間競争が無く親個体と次世代個体は強制的に入れ替わるため，親個体の形質は次世代個体を通してのみ集団に保存される．そのため，親子が同時に次世代に残り類似した形質が集団に広まることを抑制している．その一方，子個体間・変異個体間での生存競争が存在するため，良好な形質のみが集団に保存される．また，子個体は親 1 ペアから最少である 1 個体のみが生存を許されているため，類似した形質をもつ同じ両親から生成された複数の子個体が集団に同時に生存することはできない．従来までは，個体単位での多様性や集団中の適応度の分散の大きさを重視することが多かったが，本手法では，個体の持つ形質の多様性を考慮している点に特徴がある．

4. グリッド環境における GSA の実装

GSA を次のように実装することで，解品質を維持したままで計算時間を短縮する並列型 GSA が実現可能であると考えられる．非同期性を考慮し，サーバの役割分担をマスター・スレーブ型とし，世代交代を担当するマスターホスト，および，個体生成と評価を担当する複数のスレーブホストにより構成する．すなわち，2.2 節における(2)の方法を採用する．比較的計算量の大きい個体評価を並列に分散されたスレーブホストが処理するため，全体の計算時間を短縮することが可能であると考えられる．選択・淘汰はスレーブホスト内で局所的に行われるためスレーブホスト間の同期を考慮する必要がなく，非同期かつ並列に世代交代は行われる．なお，並列数は最大で集団サイズと同程度まで大きくすることが可能である．通信量に関しては，スレーブホストに送信する通信量は親個体の情報および最悪個体の評価値のみであり，また，スレーブホストからの受信は次世代個体 1 個体の情報のみである．

なお，並列型 GA は，島モデル型 GA とするのが一般的であるが，評価回数を同等とした条件下では，非並列な非島モデル型の GA と比較して，非並列な島モデル型 GA から得られる解品質は低いことを著者らは確認している（染谷・山村 (1999)）．

5. 実験

5.1 異なる通信条件における性能評価

スレーブホスト内の処理を一定時間のスリープとして簡略化した GSA を実装し，通信条件による計算時間への影響について調べる．

2 台のサーバ，PC1 および PC2 を使用する．PC1 は統計数理研究所内（東京都港区)に，PC2 は理化学研究所ゲノム科学総合研究センター内（横浜市鶴見区)に設置されている計算機である．両 PC は OBIGrid（Open Bioinformatics Grid)と呼ばれるグリッドに接続されており，本実験はこの環境上にて行われる．GSA の実装構造を図 4 に示す．GSA は Java 言語にて記述され，並列計算の処理は RMI の機能を利用する．ユーザ認証および RMI サーバの起動は Globus の機能を利用する．

マスターホストの機能は PC1 が担当するものとする．通信の遅延時間による影響を調べる

図 4.　並列型 GSA の実装構造.

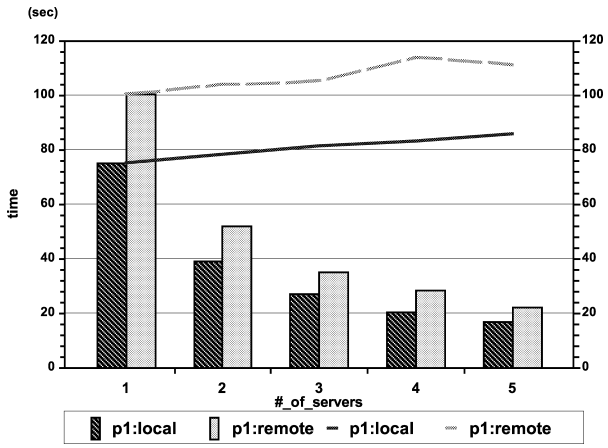

図 5.　サーバ数の変化に伴う並列型 GSA の計算時間の変化 (a).

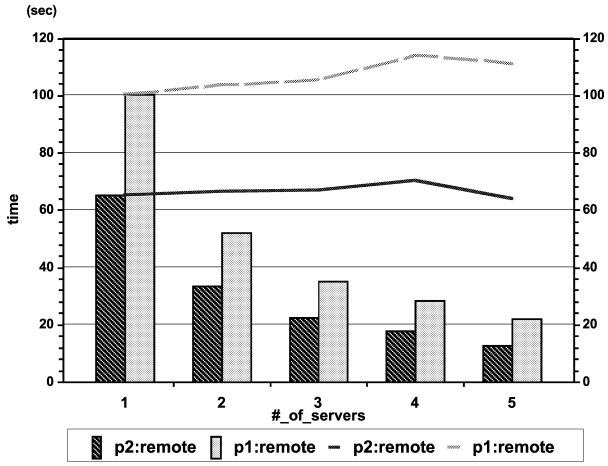

図 6.　サーバ数の変化に伴う並列型 GSA の計算時間の変化 (b).

ために，スレーブホストの機能もPC1が併用される場合とPC2が担当する場合について比較する．また，通信頻度の影響を調べるために，通信頻度の異なる2つのパラメータセットを使用し比較する．頻度の多いものは，ファミリー内の処理が0.1秒，世代数が200であり，頻度の少ないものは，ファミリー内の処理が1秒，世代数が20である．

実験結果を図5および図6に示す．図5はスレーブホストをPC1とした場合（図中のlocal）とPC2とした場合（図中のremote）を比較しており，図6は通信頻度が多い場合（図中のp1）と少ない場合（図中のp2）の比較である．グラフの横軸は使用した仮想的なサーバ数を表している．棒グラフは計算時間の実時間を示しており，折れ線グラフは「実時間 $\times$ サーバ数」である．なお，実験は比較的通信の輻輳の影響の小さい深夜から明け方の時間帯に行われた．図5から，PC2を使用した場合は計算時間が長く，通信の遅延時間による影響が確認できる．また，図6から通信頻度の影響が確認できる．図5と図6のいずれの場合もサーバ数の増大に対してほぼ比例して計算時間が短縮されており，サーバ数がより多くなった場合にも線形のオーダーでの計算時間短縮が期待される．

5.2 最適化問題における性能評価

Kauffman and Macready（1995）により提案された$\mathcal{NP}$完全な最適化問題であるNK-model問題に並列型GSAを応用しその性能を評価する．NK-model問題における評価関数を以下に示す．

$$Fitness = \frac{1}{N}\sum_{i=1}^{N} f_i\,, \tag{5.1}$$

$$f_i = f(x_i; x_{j0}, \ldots, x_{jK})\,, \qquad f(x_i) \sim U(0,1)\,.$$

Nは探索空間の次元であり，すなわち，染色体長を表す．Kは，各遺伝子が依存関係を持つ他の遺伝子の数を表す．各$x_{j0}, \ldots, x_{jK}$はそれぞれA種類の対立遺伝子を取る．NK-modelの探索空間の景観の起伏は，Kの値によって調整することができる．$K = 0$の場合は，景観は単峰性になり，変数間の依存関係はない．一方，Kが0よりも大きい場合は，景観は多峰性にな

表1. 実験パラメータ．

並列型SA	
状態系列数	30
初期温度	10,000
温度減少率	0.95
一温度あたりの状態遷移数	750

並列型GSA	
集団サイズ	30
ファミリーあたりの個体生成数	500

表 2. 最適解到達率 (%).

	並列型 GSA	並列型 SA
K=1	100	90
K=2	100	3
K=4	87	4

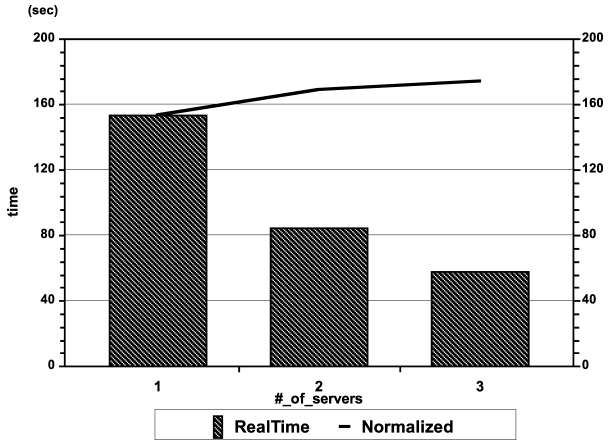

図 7. NK-model における並列型 GSA の計算時間.

りいくつかのピークを持つ．K が大きくなるに従い，ピークの数も大きくなり，変数間の依存関係も強くなる．

3 つのベンチマーク問題を作成し，並列計算機環境におけるポピュラーな最適化手法である並列型 Simulated Annealing (SA) と比較する．問題パラメータの N, A はそれぞれ，$N = 32, A = 2$ とする．K には，$K = 1, 2, 4$ を用い，異なる種類の景観における性能比較を行う．予備実験により決められた並列型 SA および並列型 GSA の各パラメータを表 1 に示す．並列型 GSA の交叉オペレータには，両親からの形質遺伝の比率が正規分布に従う一様交叉を用いた．両手法とも打ち切り評価回数を 200,000 回と同一にし，100 試行における最適解到達率を比較する．実験環境は同スペックのサーバ 3 台により構築された仮想的なグリッド環境である．

実験の結果を表 2 に，並列型 GSA の計算時間を図 7 に示す．並列型 GSA は，最適化が困難とされる多峰性の問題に対しても高い性能を示しており，また，サーバ数に対してはほぼ比例した計算時間の短縮を実現していることが確認できる．

6. まとめ

本稿では，グリッド環境に適した GA について考察し，グリッド環境に適した GA とは，1. 世代交代が局所化されている，2. 通信量が小さく通信頻度が少ない，3. 高性能である，を満たした GA であることを示した．また，適応的な探索を特徴とする GA の一手法である GSA

を概括し，グリッド環境への実装例およびその有効性を示した．今後は，大規模なグリッド環境における並列型 GSA の実現可能性およびその性能について調査する必要がある．

謝　辞

本研究を進めるにあたり，多大な協力を頂きました小長谷明彦氏(理化学研究所 ゲノム科学総合研究センター プロジェクトディレクター)および OBIGrid 参加機関の皆様に深く感謝致します．本研究の一部は，科学研究費補助金 若手研究（B）「次世代超並列計算環境上での進化型計算（課題番号 14780298）」および，電気通信普及財団 研究調査助成「広域超並列計算環境における進化型計算の実装と挙動解析」により行われました．

参 考 文 献

Baker, R., Buyya, R. and Laforenza, D. (2000). The grid: International efforts in global computing, *International Conference on Advances in Infrastructure for Electronic Business, Science,and Education on the Internet* (*SSGRR 2000*), l'Aquila, Rome, Italy.

Foster, I., Kesselman, C. and Tuecke, S. (2001). The anatomy of the grid: Enabling scalable virtual organizations, *International Journal of Supercomputer Applications*, **15**.

Kauffman, S. A. and Macready, W. G. (1995). Search strategies for applied molecular evolution, *Journal of Theoretical Biology*, **173**, 427–440.

小林重信 (1993). 遺伝的アルゴリズムの基礎と応用 [I]，オペレーションズ・リサーチ，5 月号，256–261.

Open Bioinformatics Grid. http://www.obigrid.org/

染谷博司 (2001). 探索オペレータの機能分担を考慮した進化型計算による最適化，東京工業大学，博士論文 甲第 4805 号.

染谷博司 (2003). グリッドでの最適化について，研究会「並列計算・乱数・グリッド・HPC」講演資料，統計数理研究所，東京.

染谷博司 (2004). 進化型計算による適応的探索およびグリッド環境への応用，最適化：モデリングとアルゴリズム 17，統計数理研究所共同研究リポート，No. 168, 180–190.

染谷博司，山村雅幸 (1999). 探索領域を適応的に調整する遺伝的アルゴリズムによるフロアプラン設計問題の一解法，電気学会論文誌 C, **119-C**(3), 393–403.

染谷博司，山村雅幸 (2002). 探索オペレータの機能分担を考慮した進化型計算による関数最適化，電気学会論文誌 C, **122-C**(3), 363–373.

武田伸悟，伊達進，下條真司 (2003). グリッドファイルシステム GSI-SFS, 情報処理学会研究報告(システムソフトウェアとオペレーティング・システム), **2003-42**, 97–104.

The Globus Project. http://www.globus.org/

山村雅幸, 小野貴久, 小林重信 (1992). 形質遺伝を重視した遺伝的アルゴリズムに基づく巡回セールスマン問題の解法，人工知能学会誌，**7**(6), 1049–1059.

Proceedings of the Institute of Statistical Mathematics Vol. 52, No. 2, 381–391 (2004) 391

A Genetic Algorithm Optimization on Computational Grid

Hiroshi Someya

The Institute of Statistical Mathematics

Recently, Genetic Algorithm (GA) is recognized as a useful optimization tool because of its high performance. However, a major drawback of GA is its heavy computational load. This paper deals with the application of GA to grid computing. It discusses several features of grid computing, and then introduces an implementation of a GA, *Genetic algorithm with Search area Adaptation* (GSA), on a computational grid. It is shown through numerical experiments that the performance of GSA is proportional to the number of server computers and it is superior to a parallel simulated annealing algorithm.

Key words: Genetic algorithm, grid, distributed computation, parallel computation, optimization.