ハイパフォーマンス
コンピューティング 91−14
（２００２．８．２１）

Grid 環境における評価部に個体データベースを用いた遺伝的アルゴリズムの提案

廣安知之[††] 三木光範[††]
片浦哲平[†] 谷村勇輔[†]

本研究では Grid 計算環境を想定した最適化システムの提案を行っている．提案するシステムは最適化エンジン，計算サーバ，データベースサーバから構成されている．最適化計算に必要な評価値の導出は，計算シミュレーション，データサーバに格納済みデータ，データサーバのデータからの近似値を利用することが可能である．通常，最適化計算においては複数回の試行を行うために，シミュレーション結果をデータベースに格納し，次回の試行の際に使用することで，試行が増すごとに処理速度が向上する．また，評価値を求めるのに計算コストが大きな場合，近似値を使用することで計算負荷の削減が可能である．本研究では最適化計算部に遺伝的アルゴリズムを適用したシステムを構築し，数値計算例を通じてその有効性を検討している．

Genetic Algorithm using a Population Database on the Grid

Tomoyuki Hiroyasu,[††] Mitsunori Miki,[††] Teppei Kataura[†]
and Yusuke Tanimura[†]

In this study, we proposed the optimization system in the computational GRID. The proposed system consists of the optimization engine, the calculation server, and the database server. The evaluation value of objective function can be derived from the calculation simulation, the stocked data of the data server, and the approximation value from the stocked data. Usually, the optimization operation needs the plural trials. Therefore, when the results of the calculation simulation are stocked and these values are used in the next trial, the operation speed can be increased in the next trial. At the same time, when the calculation cost of the evaluation value is very expensive, the calculation cost can be reduced by using the approximation values. In this system, the genetic algorithm is applied in the optimization engine. Through the numerical simulations, the effectiveness of the proposed system is discussed.

1. はじめに

1980 年代のワークステーションをネットワーク化することにより始まった分散処理は，LAN(Local Area Network) の高速化にともない，クラスタシステムと呼ばれる仮想的な並列計算機に進化した．近年，ネットワークのさらなる技術向上により，LAN だけでなく WAN(Wide Area Network) においても高速なネットワークが利用可能となった．このため，並列計算の分野は，LAN レベルの狭い環境だけでなく，広域の環境においても適用可能な技術的背景が整ってきた．その中で，遠隔地の計算資源を結びつけ，ひとつの巨大なコンピュータとして有効活用する Grid[1)] に関する研究が盛んに行われるようになった．Grid は，スーパーコンピュータの計算性能を超える大規模計算機，ペタバイト級の大規模データを格納できる巨大なデータベースとなる可能性を持っている．そのため，前者は計算を行うための Grid として，後者はデータを利用するための Grid として科学技術計算分野への応用が考えられている．

科学技術計算分野において，目的関数の最大値，最小値を求める最適化は重要な問題である．しかし，近年，最適化問題は大規模かつ複雑化しており十分な解を得るまでには多くの反復計算が行われるため膨大な計算量が必要となってきた．そのため，一度計算された情報を格納するデータベースを利用することで評価時間の短縮を図ることが考えられている．Grid は，最適化問題を解くための高い計算性能と，計算された情

† 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

報を格納できる膨大な量のデータベースを有した環境であるといえる．このため，Grid 上で計算とデータベースを融合させたシステムを構築できれば，最適化計算を行う有効な手段となる．本論文では，そのシステムの提案を行う．

また，本論文では，提案システムの最適化問題の解法手段として遺伝的アルゴリズム (Genetic Algorithm：GA)[2] を採用した．GA は，多点探索による大域的な探索が可能であるが，評価に対する計算量の膨大さが深刻な問題となっている．しかし，GA の一部の計算処理は同時並行に実行する事が可能[3] なことや，厳密に計算を行う必要のある解析計算などとは異なり，確率や経験に基づく計算を行うため，動的に計算手順を変えたり，部分的な計算を破棄することが可能であるため，Grid のような動的な環境に適した特性を有しているといえる．

そこで，本論文では提案するシステムの最適化計算部分に GA を適用し，システムの構築を行う．また，数値計算例を通じて提案システムが Grid 上において有効なシステムであるかの検討を行う．

2. データベースを用いた Grid 最適化システム

2.1 提案システム

提案システムを図 1 に示す．提案システムは，Optimization Engine，Function Agent，Database Server，Analysis Server，Approximation Server から構成される．それぞれの役割は以下の通りである．

- **Optimization Engine**
  Optimization Engine は，最適化手法の最適化計算部分を担うサーバである．最適化手法は決定論的手法と確率的手法に分類されるが，どちらの手法も実装することができる．
- **Function Agent**
  Function Agent は，各サーバから送られてくる情報を振り分ける役割をする．
- **Database Server**
  Database Server は，実際の実験およびシミュレーション結果を格納する．また，Database Server は，一度計算された目的関数の評価値と実際の実験によって得られた結果を格納することが可能である．Function Agent から目的関数の評価依頼があった場合には，まず Database Server で検索を行う．検索に成功した場合には，Function Agent に成功したことを伝え，目的関数値を Optimization Engine に送信するように依頼する．検索に失敗した場合には，Function Agent に失敗したことを伝え，近似を行う場合には，データベースに格納されている情報から近似に利用する情報を選択し，Function Agent に Approximation Server に送信するように依頼する．また，Function Agent から送信された目的関数値の格納を行う．
- **Analysis Server**
  Analysis Server は，Database Server が検索に失敗した場合に Function Agent から送信される情報をもとに目的関数の評価を行う．目的関数値が求められると Function Agent に送信する．
- **Approximation Server**
  Approximation Server は Database Server が検索に失敗した場合に Function Agent から送信される Database Server の情報をもとに近似を行う．近似評価値が求められると Function Agent に送信する．

2.2 近似操作

近似は各最適化手法の最適化計算部分の継続を目的とする．評価計算の膨大な問題では，関数の評価に多くの時間を要するため，それ以外の処理に待ち時間が存在する．しかし，評価計算を行っている間に，目的関数値の近似値を返し最適化計算部分の操作を継続させることでリソースを有効活用し少しでも速く最適解を得ることを試みる．

2.3 優先評価

近似によって最適化計算が継続できるようになると，評価計算の処理に比べて最適化計算が進むので，評価されずに待機状態になる情報が大量に発生する．この問題を解決するには Analysis Server に送信できる情報量を調整する必要がある．提案システムでは，待機状態となった未評価の情報の送信された時間を把握しておき，時間の情報から，最新の未評価情報を優先的に評価計算させる方法で優先評価を行っている．これによって，解探索のより進んだ情報から評価計算を行うことができるので解探索を速めることができる．

2.4 提案システムの特徴

提案システムの特徴を以下に述べる．

- データベースにはシミュレーションによって解析を行った目的関数値と実際の実験によって得られた結果の 2 種類が含まれている．検索を行う際には実験によって得られた結果を優先する．
- 近似サーバは近似を行うと有効な最適化手法において，必要な場合に利用することができる．

2.5 提案システムの利点

提案システムの利点を以下に述べる．

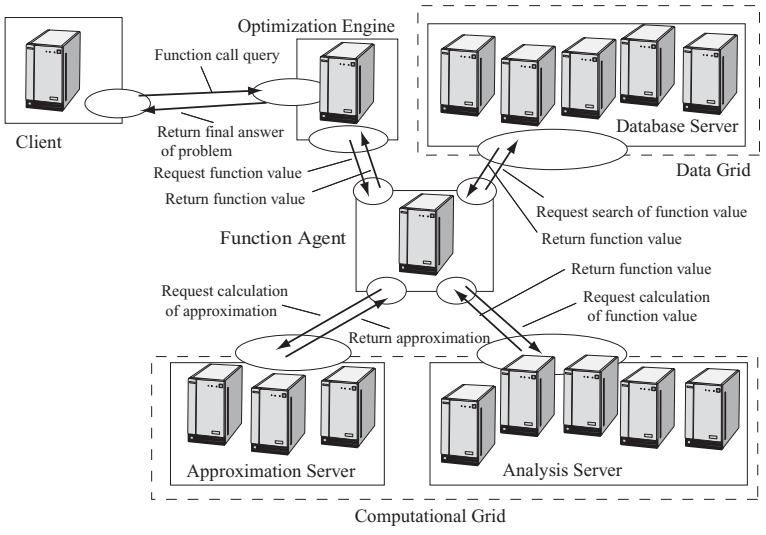

図 1 提案システム

- データベースに実際の実験結果を格納することができるので，シミュレーションでも実際の実験結果を使用することができる．また，同じ情報の重複計算を防ぐことができる．
- GA などの確率的手法は複数回の試行を行うことが一般的であるが，その場合に，データベースを利用することで情報が蓄積されていくので，試行ごとに処理時間を短縮することができる．
- Grid での使用を想定しているので，世界中で行われている同様のシミュレーションや実験結果を集積することで，より高速な最適化が実現できる．
- 計算負荷の著しく高い問題には，近似サーバを利用することで計算時間の短縮を図ることができる．
- 近似を用いた場合には，評価サーバに最新の情報を優先的に送信することで評価計算を有効に行うことができる．

3. GA を実装した提案システム

3.1 GA

GA は，生物の進化を模倣した最適化手法である．生物は 1 つの個体で表現され，個体は染色体によって特徴を持ち，染色体は遺伝子の集まりから構成される．個体には適合度が設定され，適合度の高い個体ほど対象となる問題の評価値は最適値に近くなる．そして，図 2 のように個体群に対し評価を行う．そして，選択，交叉，突然変異などの遺伝的操作を行い新たな個体群を生成する．この操作を繰り返すことで，優れた個体を作っていき，やがて最適解に到達させる方法が GA の基本的概念である．GA は計算負荷が高いことが問題だが，GA の一部の計算処理は同時並行に実行する事が可能なため，その並列性を利用して GA を改良し，より理想的な解を高速に見つけることを目標にした並列遺伝的アルゴリズム (Parallel Genetic Algorithm：PGA)[4] の研究も行われている．それらは，負荷の分散方法によって単一母集団マスタースレーブモデル (Single population master slave model)[3),5)]，粗粒度並列化モデル (Coarse grained parallel model)[6]，細粒度並列モデル (Fine grained model)[3] などのモデルに分類される．本システムは，マスタースレーブモデルに当てはまる．

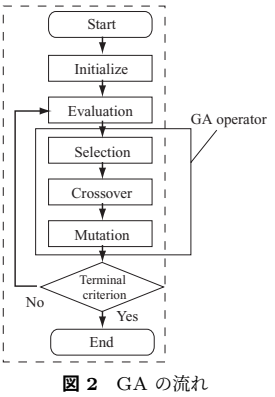

図 2 GA の流れ

3.2 GA を実装した提案システムの流れ

図 3 をもとに GA を実装した提案システムの動作手順を説明する．提案システムは以下の (1)〜(8) の処理を繰り返すことで動作する．

(1) **Initialize**

Optimization Engine が GA の母集団を作成し

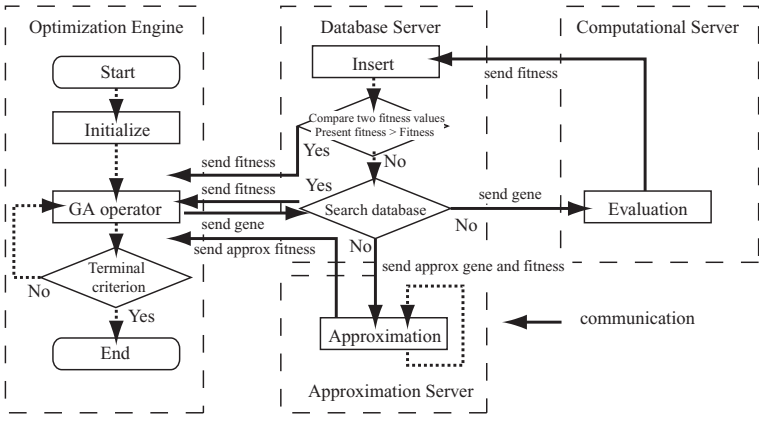

図 3 GA を実装した提案システムの流れ

個体の初期化を行う.

(2) **GA operator**
Optimization Engine が各個体に対し最適化計算部分である遺伝的操作を適用する．遺伝的操作が完了すると Database Server に染色体の情報を送信し，個体の適合度の検索を依頼する.

(3) **Search database**
Database Server が染色体の検索を行う．染色体は 0,1 の遺伝子で構成されており，検索は遺伝子のマッチングにより行う.

(4) **Approximation**
Approximation Server が Database Server は検索に失敗した個体を受け取り，何らかの方法で近似した適合度を Database Server に送信する．Database Server は，受信した近似適合度を Optimization Engine に送信する．近似の詳細については後述する.

(5) **Evaluation**
Analysis Server が Database Server から受信した染色体について評価を行う．求められた適合度を Database Server に送信する.

(6) **Insert**
Database Server が受信した適合度を対応する染色体とともに格納する.

(7) **Compare two fitness values**
Database Server が受信した適合度と，これまでに格納された適合度と受信した適合度で比較を行う．受信した適合度がさらに優秀な個体であった場合には，Optimization Engine に最も優秀な個体としてすぐに送信する.

(8) **Terminal criterion**
遺伝的操作の終了判定を行う.

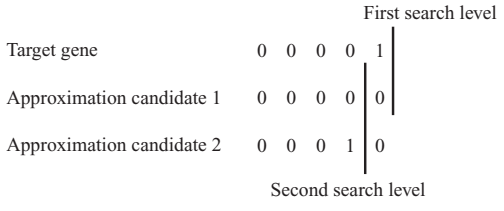

図 4 近似染色体の決定

3.3 GA における近似方法

GA における近似方法を以下に示す.

(1) 「検索対象の染色体」の検索が失敗した場合，検索に失敗した同位の逆遺伝子から検索を開始する.

(2) 同位の遺伝子の検索に失敗した場合には，1 つ上位の遺伝子を「検索対象の染色体」とハミング距離の等しい構成になるものから検索する.

(3) 検索に成功した場合は，その染色体の適合度を「近似染色体 1」の適合度とする.

図 4 の場合,「検索対象の染色体」は「00001」でその第 1 候補が「00000」，第 2 候補が「00010」となる．「近似染色体 2」も「近似染色体 1」と同様に決定する．2 つの染色体が選択されると，次に近似を行う．近似は 2 つの染色体をもとに，以下の式にしたがって行う.

$$Approx\ \ fitness = \frac{F_1 \times H_2}{H_1 + H_2} + \frac{F_2 \times H_1}{H_1 + H_2}$$

F_1 … 「近似染色体 1」の適合度
F_2 … 「近似染色体 2」の適合度
H_1 … 「近似染色体 1」とのハミング距離
H_2 … 「近似染色体 2」とのハミング距離

近似適合度は，2 つの染色体の適合度のハミング距離によって求められる．したがって，近似式は「検索対象の染色体」とハミング距離が近い染色体が見つか

るほどその適合度が反映される．

4. 数 値 実 験

4.1 One Max 問題の適用

2.5 節で説明した提案システムの利点のうち，複数試行での計算時間の短縮，重複計算の防止，近似，優先評価の有効性を確認するために数値実験を行った．対象問題には OneMax 問題を用いた．

One Max 問題とは，遺伝子の 1 の合計数がそのまま適合度となる問題である．例えば，遺伝子長が 100 の問題では，遺伝子配列はすべて 1 となっている状態が最適値であり，その場合の適合度は 100 となる．

4.2 システム概要

GA を実装したシステムは一部が 2 章で説明したシステムと以下の点で実装が異なっている．

- 実験は PC クラスタ上で行っている．
- Database Server が Function Agent と Approximation Server を兼ねている．
- Database Server は 1 台である．

また，GA の場合には，データベースに格納する情報として，適合度と適合度を求める際に必要である染色体を格納する．格納可能な染色体数は 2000 である．

4.3 実 験 内 容

データベースの効果を確認するための実験 (実験 1) として，複数試行した場合に各試行ごとに実行時間がどのように短縮されるかを調査した．また，全評価個体のうちデータベースの検索によって適合度を得ている個体の割合を求めた．加えて，近似と優先評価の効果を確認するための実験 (実験 2) として，データベースで個体の検索のみを行い検索に失敗した場合には，近似を行わずに評価値を得るまで待つシステムを構築し解が求まるまでの実行時間の比較を行った．実験 1 は，個体の遺伝子長に対するスケーラビリティを調査している．これにより，探索空間が広がるので少ない世代数では，データベースの検索が成功しなくなることが予想される．同様に，複数試行した場合，探索空間が広がることで全体的な検索成功率も下がると予想されるので，時間の短縮に影響が出ると思われる．実験 2 は問題負荷を変え 1 評価あたりの計算時間を調整することで，様々なサイズの問題を想定している．問題負荷とは，1 評価あたりの計算時間の指標であり，問題負荷の最も低い問題を 1 とした場合に，1 評価に何倍の計算時間を要するかを示している．実験に用いたパラメータを表 1 に示す．

4.4 複数試行での実行時間の短縮

データベースによって複数試行での実行時間短縮されているかを表 1 の実験 1 のパラメータを用いて調べた．実行結果を図 5 に示す．

表 1 実験 1,2 のパラメータ

	Experiment 1	Experiment 2
Population	50	50
Gene length	50, 100, 200	100
Crossover rate	1.0	1.0
Mutation rate	0.02, 0.01, 0.005	0.01
Elite poplulation	20	20
Ploblem load	1	1, 10
Trial	20	20
Terminal criterion	1000 generation	Discover optimal

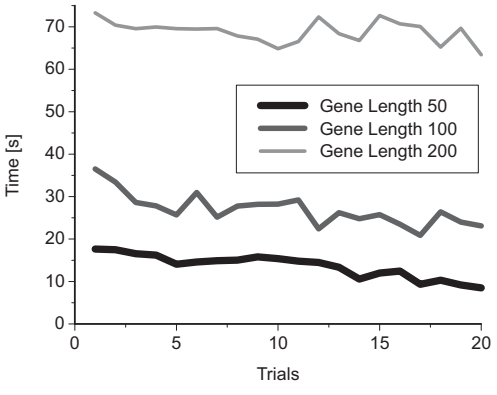

図 5 複数試行による実行時間 (実験 1)

図 5 は各試行ごとの最終的な実行時間を示している．どの条件による実行結果も全体的に右下がりになっているため，複数試行時に実行時間が短縮されていることが分かる．例えば，遺伝子長 50 の場合の実行結果は，1 試行目に 20 秒近くかかっていたのが，20 試行目では 10 秒前後にまで短縮し，およそ半分の時間で 1 試行を終えている．しかし，遺伝子長が大きくなるほど実行時間の短縮率が低くなり，さらに，各試行ごとに実行時間にばらつきがあることから，遺伝子長を大きくした場合は，十分な量の適合度が格納される必要があるということがいえる．

4.5 検索成功率の推移

全評価個体のうち検索によって適合度を得ている個体の割合を求めた．表 1 の実験 1 のパラメータを用いて，20 回試行の平均値での実行結果を図 6 に示す．

図 6 は，データベースで検索が成功した確率を示す．例えば，遺伝子長が 200 の場合には Optimization Engine で 45000 回程度の評価依頼が出された時点で最適解を得ているが，この時，検索によって適合度を得た割合は 60 %前後となっている．このことから，多くの個体がデータベースで検索に成功していることが

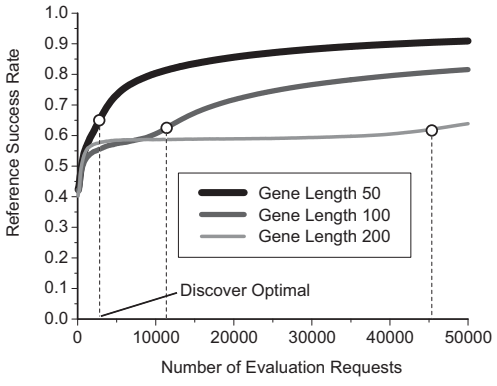

図 6 データベースの検索成功率 (実験 1)

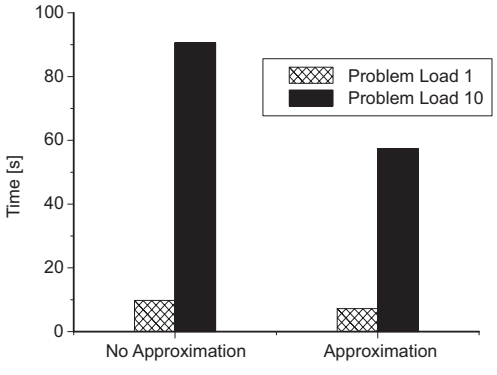

図 7 近似の有無よる実行時間 (実験 2)

分かる．また，検索時間と評価計算時間の比率を調べると全検索時間は全評価計算時間の 0.1 %であった．そのため，検索によって適合度を得る方が評価計算によって適合度を得るよりも計算時間を短縮することができるといえる．以上から，無駄な評価計算を省くためにデータベースが有効であるといえる．

4.6 近似の有無による解探索性能の違い

最後に，近似を行わずに評価計算によって適合度が求められるまで遺伝的操作を待つ手法 (近似を行わない手法) とデータベースの検索に失敗した場合に，近似を行い，評価計算待ちの個体が発生した場合に，最新の個体を優先的に評価する手法 (近似を行う手法) の 2 種類の手法で，最適解を得られるまでに要した時間を求めた．表 1 の実験 2 のパラメータを用いて，20 回試行の平均値での実行結果を図 7 に示す．計算量は，1 回の計算につき評価関数を複数回呼び出すことで人為的に大きくした．本問題の場合，問題負荷 1 は 1 回の計算につき 10 万回の評価関数を呼び出し，問題負荷 10 は 100 万回の評価関数を呼び出している．

図 7 から，近似を行わない手法と比較して近似を行う手法がよい性能を示した．問題負荷の大きな問題でその差が顕著であることから，大規模な問題においてはより優れた手法であるといえる．また，近似および最新の個体を優先的に評価することが解探索に有効なことから，Grid 環境に適したモデルであるといえる．

5. ま と め

本論文では，Grid 環境に適応した最適化システムとしてデータベースを用いたシステムを提案した．提案システムは，データベースを用い検索を行うことで計算時間の短縮を図ることや，近似を行うこで最適化計算を継続しリソースを有効活用できること，最新の情報を優先的に送信することで評価計算を有効に行うことができるなどの利点が挙げられた．提案システムの有効性を確認するために，最適化計算に GA を実装して実験を行った．また，提案システムに One Max 問題を適用することで，複数試行，検索，近似，優先個体の評価が有効であることが確認された．このことから，Grid 環境に適応した最適化システムとして有効性が示されたといえる．

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) Ian Foster, Carl Kesseleman. *The Grid : Blueprint for a New Computing Infrastructure.* Morgan Kaufmann,1998.
2) D.E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley,pp.113-120,1989.
3) Erick Cantú-Paz. *A survey of parallel genetic algorithms.* Calculateurs Paralleles,Vol.10,No.2,1998.
4) Erick Cantú-Paz and David E. Goldberg. *Predicting Speedups of Ideal Bounding Cases of Parallel Genetic Algorithms.* International Conference on Genetic Algorithms,1997.
5) D.Levine. *A parallel genetic algorithm for the set partitioning problem.* Technical Report ANL-94/23,Algonne National Laboratory,Mathematics and Computer Science Division,1994.
6) Chrisila C. Petty and Michael R. Leuze. *A Theoretical Investigation of a Parallel Genetic Algorithm.* Proc. 3rd International Conference on Genetic Algorithms,pp.398-399,1989.
7) Abramson D, Lewis, A, Peachey T, Fletcher, C. *An Automatic Design Optimization Tool and its Application to Computational Fluid Dynamics.* SuperComputing 2001.