ハイパフォーマンス
コンピューティング 91−15
（２００２．８．２１）

Grid環境における進化計算手法の検討

谷村勇輔[†] 廣安知之[††]
三木光範[††] 青井桂子[†]

本論文では，Grid 計算環境において進化計算システムを開発するために「EVOLVE/G」システムを提案する．EVOLVE/G システムは，既存の Grid RPC に基づくシステムでは構築しにくい並列モデルを構築できる．また，Grid 環境内のノードをクラスタリングし，階層的な計算モデルを記述することも可能である．これらによって，進化計算研究者が自分たちの望む自由でスケーラブルな並列モデルを記述できると考える．本論文では，進化計算の 1 つである PSA/GAc を例にとり，その Grid モデルについて検討を行った．数値実験を通して，EVOLVE/G を用いて進化計算のモデルを検討することの有効性を確認した．

Discussion on Evolutionary Computing on the Computational Grid

Yusuke Tanimura,[†] Tomoyuki Hiroyasu,[††] Mitsunori Miki[††]
and Keiko Aoi[†]

In this paper, we proposed the "EVOLVE/G" system for developers to construct evolutionary computation (EC) system on the computational Grid. The Grid RPC can construct only the master-slave model of EC. On the other hand, the EVOLVE/G can implement several types of parallel models of EC. In addition, the EVOLVE/G has a function to cluster the computational nodes on the Grid. This function enables for developer to describe the hierarchically computational model. The researchers can try to implement several models of EC freely. In this paper, the Grid calculation model of Parallel Simulated Annealing using the Genetic Crossover (PSA/GAc) is developed using the EVOLVE/G. The numerical experiment has been performed in the real Grid environment. Through this experiment, the effectiveness of EVOLVE/G is discussed. Consequently, it is shown that the examination of the calculation model using the EVOLVE/G is effective.

1. はじめに

Grid 技術は広域に存在する計算資源や情報資源を結びつけ，分散・並列計算を行うことを可能にする．そのため，近年増大しつつある大規模計算の要求を解決する技術として期待されている[2]．特に，近年の Grid 技術は構築・利用方法に関する研究が実を結びつつあり，いくつもの試験環境が整備され，実際のアプリケーションを動かすことができる段階となっている．Grid 環境において，アプリケーション開発に専念するためにいくつかミドルウェアが開発されているが，その代表が Grid RPC に基づくシステムである[6]．Grid RPC に基づくシステムでは，サーバ側にあらかじめ用意しておいた数値計算ライブラリなどをクライアント側から容易に利用できる仕組みを提供する．

我々の研究目的は，Grid 計算環境において進化計算を用いた最適化手法を開発するツールを構築することである．構造物最適化問題やジョブショップスケジューリング問題，タンパク質の立体構造予測問題のような最適化問題を解くためには，膨大な計算が必要であり，そのために多くの計算資源が必要となる．進化計算の特徴の 1 つに多点探索がある．この特徴を利用することで，進化計算はいくつもの並列モデルを構築可能である．遺伝的アルゴリズム（Genetic Algorithms：GA）[3] を例にとると，マスタースレーブモデル[1]，島モデル[8]，セルラーモデル[5] といった並列モデルが考えられている．

進化計算の並列モデルには並列実行による処理の高速化だけでなく，並列処理による解探索性能の向上も期待されることが多い．そこで，Grid 計算環境においても，進化計算の研究者は様々な並列モデルを実際に試し，検討を行いたいのである．しかしながら，Grid

† 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

RPC に基づくシステムを用いて構築される並列モデルは，多くの場合，マスタースレーブモデルに限定される．それ以外の分散モデルを構築するためは，低レベルな通信関数を利用する必要があり，それは進化計算の研究者にとって大きな負担となっている．

本論文では，Grid 計算環境における進化計算システムを開発するための新しいツールを提案し，それを「EVOLVE/G」と呼ぶ．EVOLVE/G を用いて，進化計算のモデルを自由に構築することができる．同時に EVOLVE/G は，Grid 環境の資源を，性能やネットワーク負荷などを監視しいくつかのグループに分類する機能を有する．開発者は分類を行うためのルールを定義するだけでよい．

本論文では，PSA/GAc（Parallel Simulated Annealing using Genetic Crossover）[4] を用いて，その Grid 計算モデルを実装する．PSA/GAc はタンパク質の立体構造予測問題に適用される進化的最適化手法である．アプリケーション開発と実験を通して，EVOLVE/G が進化計算の Grid 計算モデルを検討するのに有効なツールであることを確認する．

2. GOCA

Grid 計算環境を効率的に利用できるアプリケーションは，一般的にいくつかの共通の性質をもっている．本論文では，これを GOCA（Grid Oriented Computing Application）と呼び，以下のように定義する．

- 仕事は複数に分割できる．
- 分割された仕事は互いに依存関係が少ないか，独立に実行可能である．
- あるいは，分割された仕事は並列実行可能である．
- 仕事間は頻繁な通信を必要としないか，全く通信を行う必要がない．
- ある資源が突然利用できなくなっても，低コストで仕事を継続できる．
- 新しい資源が突然加わっても，それを有効に利用できる．

進化計算は，上記の GOCA の性質の多くを満たすアプリケーションである．しかしながら，既存の Grid RPC では GOCA の特徴を最大限に引き出すことは難しい．そこで本研究においては，GOCA の特徴を引き出せる Grid のミドルウェアとして EVOLVE/G を提案する．

3. EVOLVE/G システム

EVOLVE/G システムは，GOCA を満たす様々な進化計算を容易に実装可能にする Grid ミドルウェアである．図 **1** に基本アーキテクチャを示す．

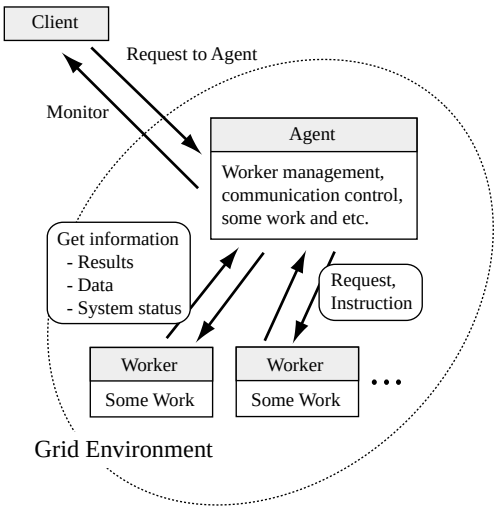

図 **1** EVOLVE/G システムの基本アーキテクチャ

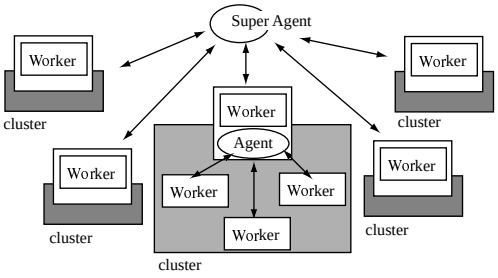

図 **2** EVOLVE/G システムの拡張アーキテクチャ

EVOLVE/G は C と Java 環境で記述され，Globus Toolkit を利用する．EVOLVE/G は Worker，Agent のコンポーネントの組み合わせからなり，それらを実装するインタフェースとして，アプリケーション開発者に API を提供する．API は MPI の Send()，Recv() 関数に似ており，各コンポーネント間のデータ通信を規定することができる．EVOLVE/G の Agent は，Worker の起動や停止，一定時間毎に Worker をチェックし，Worker に対して命令を与えたり，Worker から情報を取得したり，Worker 間の通信を仲介する役割をもつ．Worker は Agent の管理下において，コアな計算を処理する．

EVOLVE/G は基本的な Agent，Worker の関係を拡張して，これらを階層構造にすることができる．これは最上位の Agent が，Worker をクラスタリングすることで行う．クラスタリングされた場合，上位のクラスタで Worker の役割を果たしていたものが，下位のクラスタでは Agent の役割を果たすことになる．図 **2** にそれを示す．

クラスタリングは，アプリケーション開発者が決めたルールに従って行われる．例えば，各ノードの性能や負荷状況，ネットワークのスループット，ノードが所属するサイトなどを基準にした分類方法が考えられる．本研究では，クラスタリングは自動かつ動的に行われるよう計画している．しかしながら，本論文の実験においては手動でクラスタリングを行う．

4. 遺伝的交叉を用いた並列シミュレーテッドアニーリング

遺伝的交叉を用いた並列シミュレーテッドアニーリング（Parallel Simulated Annealing using Genetic Crossover：PSA/GAc）[4] は，並列に複数実行している SA の解伝達時に，GA のオペレータである遺伝的交叉を用いたハイブリッドアルゴリズムである．

一般に，SA と GA はそれぞれ次のような特徴をもつといわれている．SA は，高温で熔融状態にある物質を徐々に冷却することにより欠陥の少ない結晶などの低エネルギー状態を得る物理プロセスを，計算機上で模倣することにより最適化問題を解こうとする汎用近似解法である．現在の状態から次状態を生成する生成処理，現在の状態から次状態へ遷移するかどうかを判定する受理判定，現在の温度から次の温度を生成するクーリングの 3 つの過程を繰り返しながら，探索を行う．すなわち SA は現在の状態しか保存せずに，その近傍に重点をおいた解探索を行うことになる．

一方，GA は環境に適応した個体のみが次の世代に生き残ることができるという，生物の進化と淘汰を模倣した工学的アルゴリズムである．GA では複数の個体を解候補としてもつため，大域的な探索を行うことができる．しかしながら，2 つの探索点を組み合わせて次状態の解候補を生成する交叉オペレータでは，うまく近傍の探索が行えないことが多い．

PSA/GAc はこれらの長所を生かすよう工夫されたアルゴリズムであり，PSA を用いて局所探索を行い，GA の交叉オペレータを用いて大域的な探索を行うことが可能である．

PSA/GAc の探索手順を図 **3** に示す．

step1 初期解を生成し，複数ある探索点が並列に SA の処理を行う．

step2 アニーリングが一定期間 d に達すると，並列に実行中の SA からランダムにペアを作成する．

step3 ペアを組む 2 つの個体を親として，交叉点が 1 の設計変数間交叉を行い，2 個体の子を生成する．

step4 各ペアにおいて，親個体と子個体を合わせた 4 個体のうち，評価の高い 2 個体を選択する．

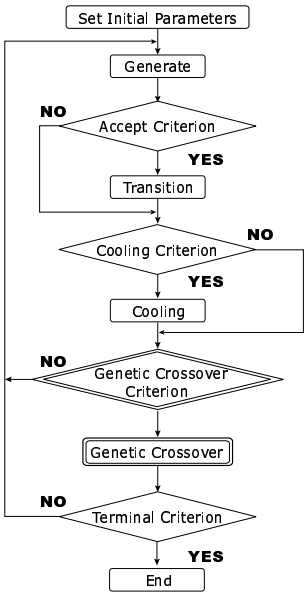

図 **3** PSA/GAc の計算の流れ

step5 選択された個体をもとに一定期間 d のアニーリングを行う．

step6 終了条件を満たすまで，step2〜step5 の処理を繰り返す．

5. EVOLVE/G を用いた PSA/GAc の論理モデルの実装

本論文では，EVOLVE/G を用いて PSA/GAc の Grid 計算モデルを検討する．検討するモデルとして，Basic モデルと Hybrid モデルを考える．Basic モデルは 1 つの Agent により管理されたモデルであり，Hybrid モデルは 2 種類の Agent によって 2 階層に管理されたモデルである．

5.1 Basic モデル

ごく一般的なモデルとして，PSA/GAc を単純に行うモデルが考えられる．すなわち，複数の SA を独立に実行し，ある一定ステップ数の後，ランダムに 2 つのプロセスを選び，新しい探索点を遺伝的交叉によって生成する．その後，次の交叉周期まで SA を実行する．本モデルを EVOLVE/G を用いて実行する場合，図 **4** に示すように，各 Worker において逐次的な SA を実行させ，唯一の Agent において遺伝的交叉を行うことにする．ただし，1 つの Worker において 1 つの SA を実行するより，複数個の SA を実行させ，その中で最も良い探索点を遺伝的交叉に参加させるといった手法が良いとされているので，これを導入する．

Worker はユーザが指定するステップ数まで各 SA

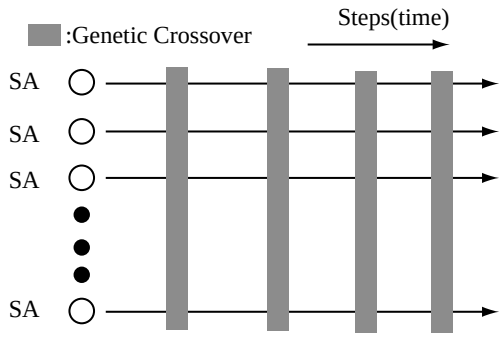

図 4 Basic モデル

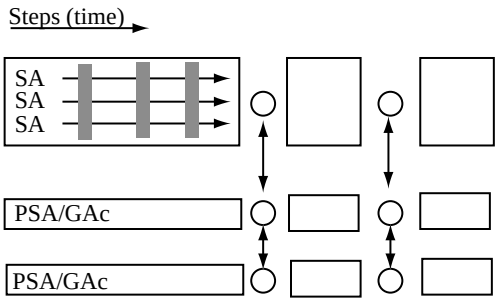

図 5 Hybrid モデル

を行い，それが終了すると最良の探索点（交叉を行う個体）をファイルに書き出し，待機状態に入る．そして Agent からのチェックを受け，交叉個体が Agent に送られる．Agent で交叉・選択が終了し，生き残った個体が送られてくることで，遺伝的交叉の操作を完了する．その後，次の交叉周期まで各 SA を行う．

Agent はユーザが指定する一定間隔毎に各 Worker のチェックを行い，Worker 間での遺伝的交叉を実現する．しかしながら，Grid 環境においては各 Worker の処理能力は均一でなく，障害によって連絡がとれなくなる Worker が出てくることが予想される．そこで Agent では，チェックの際に交叉周期に到達し，かつ通信を行える状態の Worker から取得した個体群でペアを作成し，交叉を行うことにする．このようなモデルを Basic モデルと呼ぶ．

5.2 Hybrid モデル

もう 1 つの論理モデルとして，Hybrid モデルを考える．本モデルでは複数の PSA/GAc を行う．図 **5** に示すように，各 PSA/GAc のプロセス間においては，最良の探索点を交換する操作を行う．

EVOLVE/G を用いてシステムを構築する場合，これは 2 階層のモデルとなる．本論文では，区別のために上位クラスタの Agent を Super Agent と呼ぶ．各 Agent は Worker を管理し，Worker において複数の SA を実行させる．さらに Worker から最良の探索点を集め，Agent 自身において遺伝的交叉を実行し，結果として生成された新しい探索点を Worker に戻す．Super Agent は Agent を定期的に巡回し，Agent が保存している最良の探索点を集め，Agent 間でそれらを交換するのを仲介する．PSA/GAc を行うクラスタは，Super Agent のクラスタリングによって決定される．本論文ではこれを手動で行い，ノードの所属するサイトを基準にした分類を行うことにした．本実験では 3 つの PC クラスタからなる Grid 環境を利用するため，

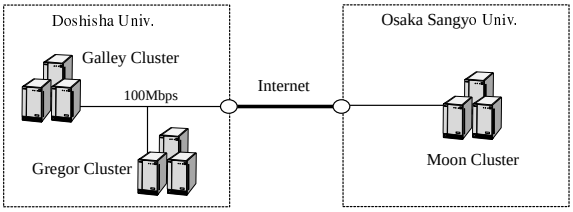

図 6 実験環境

3 つのクラスタに分かれて，それぞれが PSA/GAc を実行することになる．

6. 実験結果

6.1 実験環境

実験には，同志社大学知能情報センターに設置されている 2 つの PC クラスタと大阪産業大学に設置されている PC クラスタを用いた．各 PC クラスタのスペックは表 **1** の通りであり，OS としては Linux を用いている．それぞれのクラスタのゲートウェイマシンには Globus と MPICH がインストールされており，全てのノードで Java 実行環境が利用できる．図 **6** に示す通り，同じ建物にある Galley と Gregor は 100Mbps のネットワークで結ばれているのに対し，Galley と Moon あるいは Gregor と Moon はインターネットを介して接続されている．実験前後のゲートウェイマシン間のネットワーク・スループットを表 **2** に示す．

以下に示す実験においては，Galley のゲートウェイマシンにおいて Agent を実行し，Galley の他 8 ノードと Gregor の 12 ノード，そして Moon の 4 ノードを用いて計 24 個の Worker を実行した．

6.2 タンパク質の立体構造予測問題

本実験では，岡本らの提案するエネルギー関数を用いて，それを最小化することによりタンパク質の立体構造を予測するという問題を対象問題として用いた．対象とした Met-enkephalin は，Tyr-Gly-Gly-Phe-Met の 5 残基からなるごく小規模のタンパク質である．こ

表 1 用いた PC クラスタ

Site	Cluster name	System
Doshisha Univ.	Galley	Pentium III (1.1GHz), 1CPU Pentium III (850MHz), 8CPU
Doshisha Univ.	Gregor	Pentium III (1GHz), 64CPU
Osaka Sangyo Univ.	Moon	Pentium 4 (1.7GHz), 8CPU
Installed Software: Globus Toolkit 1.1.4, Sun JDK 1.4.0, MPICH 1.2.3		

表 2 スループットの測定

Network	Thruput	
	Message size 1KB	Message size 1MB
Galley–Gregor	10.95 [MB/sec]	11.21 [MB/sec]
Galley–Moon	107.7 [KB/sec]	92.12 [KB/sec]

表 3 エネルギー計算に要する時間

Cluster	1 step [sec]	192 step [sec]
Galley	0.215	41.4
Gregor	0.173	33.2
Moon	0.109	20.9

表 4 PSA/GAc のパラメータ設定

Number of SAs on each node	16
Initial temperature	2.0 (1000K)
Last temperature	0.10 (50K)
Crossover cycle	192 step
Range size	$180° \rightarrow (180 \times 0.3)°$

れは ECEPP/2 エネルギー関数に基づいた気相中において $E \leq -11kcal/mol$ の領域で最小エネルギー構造をしている[7]．本実験では，Met-enkephalin の主鎖における 10 個の二面角 $\phi_1, \psi_1 \sim \phi_5, \psi_5$ と側鎖における 9 個の二面角 $\chi_1^1 \sim \chi_5^4$ をそれぞれ設計変数とした．

本問題を逐次 SA で解いた時の計算時間を表 **3** に示す．これらはクラスタの各ノード毎に異なっている．表 3 は，16 個の逐次 SA を 1 ステップ実行した結果と 192 ステップ実行した結果を示している．

6.3 実験結果

実験で用いた PSA/GAc のパラメータは表 **4** に示す通りである．Basic モデルでは 24 ノードを用いて PSA/GAc が行われる．Hybrid モデルでは，それぞれ 8 ノード，12 ノード，4 ノードを用いた PSA/GAc が行われる．

まず初めに，Basic モデルの解探索の様子を図 **7** に示す．これは各チェックポイントにおいて，Agent が Worker から集めた最良の探索点のステップ数とそのエネルギー値を示している．グラフ中の値は，横軸に対してステップ数，エネルギー値ともに振動している．すなわち，性能の良いノードで実行されている Worker グループと性能の悪いノードで実行されている Worker のグループに分かれて探索が進行していると考えられる．そこで，Worker グループ全体をクラスタリングした Hybrid モデルを用いることで，より効率的な探索が行えるのではないかと予想される．

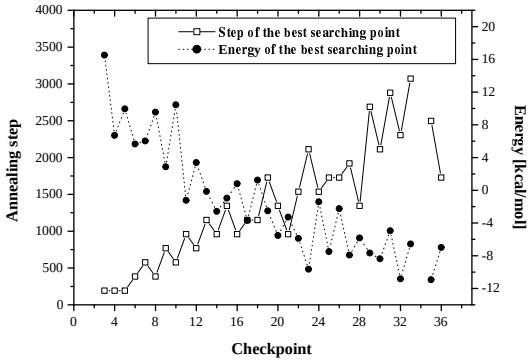

図 7 最良解のエネルギー値とステップ数の推移

図 **8**，図 **9** に Basic モデルと Hybrid モデルの比較結果を示す．図 8 は横軸がステップ数となっている．図の値は Agent が取得した情報を示しており，同じステップ数でも異なるチェックポイントで取得された探索点が存在する．特に，Basic モデルでは Worker 全体で同期がとれることは難しいため，図のような分布になっている．図より，Basic モデルに比べて Hybrid モデルが効率的に探索しているのが分かる．

図 9 は横軸が経過時間となっている．Basic モデルでは，Agent が探索の早い Worker から情報を取得する時と探索の遅い Worker から情報を取得する時があるため，最良の探索点が良くなったり悪くなったりしている．図より，Basic モデルに比べて Hybrid モデルが効率的に探索しているのが分かる．

6.4 考察

Hybrid モデルの利点としては 2 つ考えられる．1 つ目は，アプリケーションがそもそもスケールしない場合に有効である．並列アルゴリズムとしてスケーラビリティが得られないのであれば，いくつかの並列実行をパラメータを変えて行う方がずっと効率が良い．PSA/GAc を例にとると，24 並列の実行よりも 8 並列

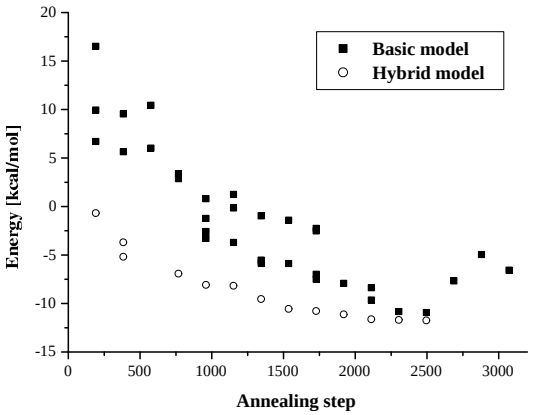

図 8 Basic モデルと Hybrid モデルの比較（横軸：ステップ数）

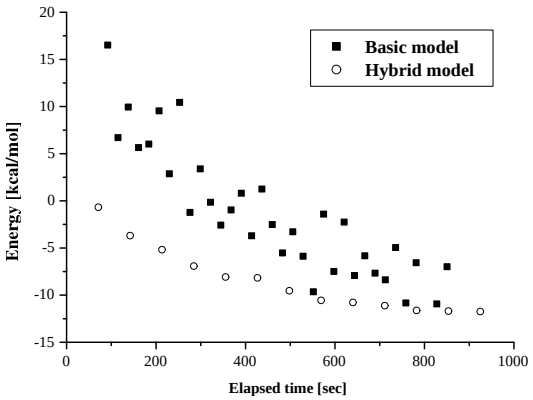

図 9 Basic モデルと Hybrid モデルの比較（横軸：時間）

や 12 並列の実行が良い探索を行うかもしれない．これは PSA/GAc の今後の研究課題である．また進化計算においては，単にパラメトリックサーチを行うよりも，各探索で何らかの情報交換を行いながら，お互いに適切なパラメータチューニングを行うといったことも可能である．

2 つ目は，通信モデルを階層的にすることで，パフォーマンスのスケーラビリティが得られることである．本論文で提案している Hybrid モデルでは，下位のクラスタ内ではプロセスにまたがって遺伝的交叉を行うなど頻繁な通信を行っている．しかし，上位のクラスタにおいては時折，探索点の交換を行うだけでほとんど通信がない．ネットワーク的に近いノードを集めたクラスタリング，あるいはクラスタシステムをそのまま利用して，各クラスタ内においては比較的密な通信を行い，クラスタ間では粗な通信を行うモデルは有効であると考える．

7. ま と め

本論文では，進化計算開発者のための Grid ツールとして EVOLVE/G システムを提案した．EVOLVE/G は進化計算の論理モデルを自由に構築できる．また，Grid 環境をクラスタリングし，ツリートポロジを形成することでスケーラブルな計算環境を構築できる．EVOLVE/G を用いて，進化計算の 1 つである PSA/GAc の Grid モデルを検討した．2 つの Grid モデルを考案し，検討を行った結果，クラスタリングを行う階層的な計算モデルが良い結果を示した．これらを通して，進化計算のようなアプリケーションの Grid モデルの構築および検討を行う際に，EVOLVE/G が有効であることを示すことができた．

謝辞 本研究を進めるにあたり，大阪産業大学の小板隆浩講師に計算資源の提供と貴重なご意見を頂きました．深く感謝いたしたいと思います．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) 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.
2) I. Foster and C. Kesseleman. *The Grid: Blueprint for a New Computing Infrastructure.* Morgan Kaufmann, 1998.
3) D.E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, 1989.
4) T. Hiroyasu, M. Miki, and M. Ogura. Parallel Simulated Annealing using Genetic Crossover. In *Proc. of the IASTED PDCS 2000*, pp. 139–144, 2000.
5) M.Gorges-Schleuter. ASPARAGOS: an asynchronous parallel genetic optimization strategy. In *Proc. of the 3rd ICGA*, pp. 422–428, 1989.
6) 中田秀基, 田中良夫, 松岡聡, 関口智嗣. Grid rpc システムの API の提案. 情報処理学会研究報告 HPC 研究会, 第 2001 巻, pp. 37–42, 2001.
7) Y. Okamoto, T. Kikuchi, and H. Kawai. Prediction of Law-Energy Structures of Met-Enkephalin by Monte Carlo Simulated Annealing. *CHEMISTRY LETTERS*, pp. 1275–1278, 1992.
8) Reiko Tanese. Parallel Genetic Algorithms for A Hypercube. In *Proc. of the 2nd ICGA*, pp. 177–183, 1987.