アルゴリズム 28－6
（1992．7．17）

並列GAにおける交換アルゴリズムの改良とその評価

棟朝雅晴　高井昌彰　佐藤義治

北海道大学工学部

遺伝子集団分割による並列遺伝的アルゴリズムにおける遺伝子交換アルゴリズムの改良とその評価を行なった結果について述べる。遺伝子集団の中に含まれるスキーマ数の減少を、その適合度の分布の標準偏差を計算することで間接的に評価し、遺伝子交換を行なう基準とすることで、遺伝子集団の中に含まれているスキーマの数の急速な減少を防ぎ、遺伝的アルゴリズムのスキーマ処理効率を維持する。そのために標準偏差と遺伝子集団内に含まれるスキーマの数について、Sigma-Hypothesis と呼ばれる仮説を立て、理論的解析とシミュレーション実験によりこれを検証した。さらに、この仮説に基づく遺伝子交換アルゴリズムについて、その有効性を実験により確認した。

An Efficient Exchange Algorithm for Parallel GAs and Its Evaluation

Masaharu Munetomo, Yoshiaki Takai, and Yoshiharu Sato

Information and Graphics Sciences, Faculty of Engineering, Hokkaido University
North 13 West 8, Kita-Ku, Sapporo 060, Japan

Improvement of exchange algorithms on parallel genetic algorithms and their evaluation are presented in this paper. To avoid rapid decrease of the number of schemata in a population and maintain efficient schema processing, the algorithm presented estimates indirectly the decrease of the number of schemata by using standard deviation of fitness distribution. We propose *sigma-hypothesis* and discuss its validity through some theoretical and empirical results. A parallel genetic algorithm based on the hypothesis is presented and some experiments are made to confirm its efficiency.

1 はじめに

遺伝的アルゴリズム (以下 GA と略す) は生物の進化と遺伝子の挙動にヒントを得た最適化アルゴリズムであり、広い範囲の組合せ最適化問題に対して比較的安定した近似最適解を得ることができる [3]。GA で大規模かつ複雑な問題を解くためには、より多くの遺伝子 (個体) が必要である。そのため、多数の遺伝子から成る集団を高速に処理することを目指した GA の並列化に関する研究がこれまで多くなされてきた。その例として、個々の遺伝子を近傍構造が定義されたある空間上に (例えば 2 次元 5 近傍格子上に) 配置し、細粒度の超並列処理を目指したもの [1][6]、遺伝子集団を中規模の部分集団に分割し、その部分集団に対して独立に中粒度の並列処理を達成するもの [7]、ハイパーキューブ型計算機へのインプリメントに関する研究 [8] などが主なものとしてあげられる。

これらのうち、並列 GA を実現する上で最も自然なアプローチは、遺伝子集団の分割による方法であると考えられる。この方法においては、遺伝子集団がいくつかの部分集団に分割され、それぞれが並列計算機のプロセッサへ割り当てられる。分割された部分集団間では、ある基準に従って定期的に遺伝子の交換を行ない、各プロセッサで独立に GA が実行される。これにより、集団分割を行なわない単純な GA と同等あるいはそれ以上の質を持った近似最適解をより短時間に求めようとするものである。

しかし、集団分割に基づいた並列 GA に関してこれまで行なわれた研究では、部分集団の間で遺伝子を交換する基準が明確に定められておらず、多くの場合全くランダムに遺伝子交換が起動されている。これは、並列アルゴリズムの性能を理論的に解析することは容易にするが、実際には無駄なプロセッサ間通信を多く引き起こすことになり、並列処理の効率を高めるという観点からは改善の余地がある。

そこで本研究では、遺伝子集団分割に基づく並列 GA において、その性能を決定する要因となっている遺伝子交換アルゴリズムについてある基準を設け、それに従った交換アルゴリズムを実現することで、並列 GA のさらなる性能向上を試みる。そのために、まず遺伝子集団の適合度の分布と遺伝子集団内の遺伝子の多様性との関係について言及する Sigma-Hypothesis を新たに提案する。続いて、これに基づいた遺伝子交換アルゴリズム Sigma-Exchange を実現し、その最適化性能をいくつかのシミュレーション実験により確認する。

2 遺伝的アルゴリズム

GA は適合度関数 (fitness function) の最大化問題を近似的に解くアルゴリズムである。最適化を行なうために、適合度関数のすべての変数は、アルファベット (alphabet) から構成される文字列である遺伝子 (string) の形式にコーディングされる。すなわち 1 個の遺伝子 (＝1 個体) は定義域内のある 1 点を示している。次に、その遺伝子を重複を許してランダムにいくつか集めた集団 (population) を構築し、これに対して以下の基本 3 操作を繰り返し適用し、より高い適合度をもった遺伝子集団に収束させる。

selection 遺伝子の適合度に応じた期待値で次世代におけるその遺伝子の個数を定める。つまり、適合度の高い遺伝子の数を増やし、逆に適合度の低い遺伝子の数を減少させる。

crossover 遺伝子を交叉させる。つまり、ある一定割合の遺伝子をペアにして、そのペアの間で文字列の部分列を相互交換する。

mutation 遺伝子を突然変異させる。つまり、ある確率で文字列中のある文字を別の文字に変化させる。

GA を特徴づける重要な概念としてスキーマ (schema) がある。これは遺伝子を構成する文字と、*(don't care symbol) からなる文字列として定義され、可能な文字列全体によって構成される集合の部分集合を表している。例えば $H = 001*01*$ というスキーマは、{0010010, 0010011, 0011010, 0011011} という遺伝子の集合に対応している。GA の効率はこのスキーマの処理効率によって決定される [7]。また、スキーマの長さを最も左に位置する文字 (*以外) と最も右に位置する文字の間の距離で定義する。例えばスキーマ $H = **10*11*$ の長さは 4 である。[3] では、この長さを *defining length* と呼んでいる。

Schema Theorem[5] により、次の *Building-block hypothesis*[3] を満たす場合に GA の特徴を生かした効率的な最適化処理が可能であることが明らかにされている。

Hypothesis 1 (Building-block hypothesis) *Building-block* と呼ばれる短くかつ高い適合度を持つ *(*そのスキーマを含む遺伝子全体の適合度の平均が高い*)* スキーマを組合せることで最適解を求めることができる。

当然のことながら、全ての問題やコーディングに対してこの仮説が常に正しいとは限らない。逆に言うと、この仮説を満たすように遺伝子のコーディングを工夫しなければならないという指針をこの仮説は表している。

3 並列遺伝的アルゴリズム

遺伝子集団分割による並列 GA は、遺伝子集団をいくつかの部分集団 (subpopulation) に分割し、それらを並列計算機のプロセッサに割り当て、独立に GA を実行する方法である。この際、部分集団間で頻繁に遺伝子を交換することで、各部分集団における遺伝子の急速な一様化が防がれている。したがってこの方法では交換をどのようにして行なうかによってその性能が決定される。遺伝子交換の方法を決定する基準として以下の 3 条件を考えることができる。

- いつ、どのような条件で交換を起動するか
- 交換すべき遺伝子の候補をどのように選ぶか
- 交換相手となる部分集団をどのように選ぶか

[7] ではこれら 3 条件をすべてランダムと仮定して、理論的解析と数値実験の結果を示している。この方法をここでは Random-Exchange と呼ぶ。Random-Exchange に基づく並列 GA (Appendix-A 参照) では、分割された遺伝子集団毎に並列に GA を実行し (5–9 行目)、ある確率で遺伝子の交換を起動する (8 行目)。*Prob_of_exchange* は交換を行なう確率パラメータである。交換相手となる部分集団と交換対象となる遺伝子も同様にランダムに選択され、遺伝子交換が実行される (9–11 行目)。

本研究では交換の起動条件を明確に与えることを目的とする。そのために、スキーマの数を近似する尺度として各部分集団で容易に計算できる適合度分布の標準偏差を採用することを考える。しかし、スキーマの数と適合度分布の標準偏差に直接的な依存関係はなく、コーディングや適合度関数の選び方に大きく依存する問題点がある。そこで、先に紹介した Building-block 仮説のように、ある仮説を立てそれがどの程度当てはまるか、実際の問題について実験を行うことでその仮説の妥当性を検証する。この仮説を *Sigma-Hypothesis* と呼ぶ。これは遺伝子集団の適合度分布の標準偏差とその中に含まれるスキーマの数について言及したもので、以下でその詳細を述べる。

4 Sigma-Hypothesis

効率的な遺伝子交換をどのように行うかについては、スキーマの処理効率、特に GA の本質的な操作である crossover の効率をどのように向上させるかということが重要である。遺伝子集団を分割すると、各部分集団に含まれる遺伝子の数が少なくなり、確率的揺らぎ (genetic drift) により適合度の小さい遺伝子が selection によって消滅しやすくなる。結果として必要以上に収束が速くなり、遺伝子の多様性が失われやすくなる。遺伝子の多様性が少なくなると、crossover によって新しいスキーマが生成されにくくなり、GA のスキーマ処理効率が著しく低下することになる。

まず、スキーマ H を含む遺伝子が、H を含まない遺伝子と crossover する確率 P_h は以下で与えられる。

$$P_h = 1 - \frac{m(H) - 1}{N - 1} \tag{1}$$

ここで $m(H)$ は遺伝子集団中でスキーマ H を含む遺伝子の数、N は遺伝子集団に含まれる遺伝子の総数である。$m(H)$ の平均を $\bar{m}$ とすると、これは次式で近似できる。

$$\bar{m} \simeq \frac{N2^l}{S} \tag{2}$$

ここで、l は遺伝子の長さ、S は遺伝子集団に含まれるスキーマの総数である。従って、スキーマの

適合度	遺伝子数	次世代における遺伝子数
f_1	k_1	$k_1' = k_1 f_1/\bar{f}$
f_2	k_2	$k_2' = k_2 f_2/\bar{f}$
$\vdots$	$\vdots$	$\vdots$
f_n	k_n	$k_n' = k_n f_n/\bar{f}$

表 1: selection の前後における遺伝子数の推移

総数Sが減少すると$\bar{m}$が増加しP_hが平均として減少するので、結果として有効な crossover が行なわれる確率が減少することになる。

この crossover 効率の低下を避けるために、スキーマの多様性、すなわち遺伝子集団内に含まれる相異なるスキーマの数を何らかの形で調べて、それが小さくなった時に他の部分集団と遺伝子の交換を行う方法が考えられる。ところがスキーマの多様性を直接数え上げることにより観測する方法は結局全数探索であり、指数オーダーの計算量がかかる。これは極めて効率が悪いだけでなく、遺伝子集団に対する操作を通じて間接的にスキーマを処理するという GA の基本的方針に反する。

そこで、我々は遺伝子集団の適合度分布の標準偏差に着目し、それにより間接的に遺伝子の多様性の減少、すなわち遺伝子集団に含まれているスキーマの数の減少を評価するアプローチを取ることとした。これは以下に示す Sigma-Hypothesis に基づいている。

Hypothesis 2 (Sigma-Hypothesis) 遺伝子集団の適合度分布の標準偏差が減少したならば、その中に含まれている相異なるスキーマの数も減少している。

これはそれぞれの値の大きさ自体については言及しておらず、その減少が同時に起こるということを示している。この仮説がどの程度正しいかについて、遺伝的アルゴリズムの selection 操作に関する簡単な解析を行なった結果について説明する。

表 1 はある適合度を持つ遺伝子の数の変化を示したものである。左の列は適合度の値、中央の列はその適合度を持つ遺伝子の数である。selection は適合度の大きさに比例した期待値で次の世代の遺伝子の数を決定するので、適合度 f_i をもつ遺伝子の数は、もとの数に $f_i/\bar{f}$ を掛けたものとなる(表右の列)。

従って、一度 selection を行なった後の適合度の平均$\bar{f}'$は以下のようになる。

$$\begin{aligned}\bar{f}' &= \frac{1}{n}\sum_{i=1}^{n} k_i' f_i \\ &= \frac{1}{n}\sum_{i=1}^{n} \frac{k_i f_i^2}{\bar{f}} \\ &= \frac{\bar{f}^2}{\bar{f}}. \end{aligned} \tag{3}$$

ここで、$\bar{f}, \bar{f}^2$はそれぞれ selection を行なう前の適合度の平均値、2 乗平均値である。$\bar{f}^2 \geq (\bar{f})^2$が常に成立するので、$\bar{f}' \geq \bar{f}$となり、適合度の平均は selection によって単調減少する。これは、selection の定義より明らかである。この結果を利用して、適合度の分布の標準偏差について調べる。selection を行なう前と後の遺伝子の適合度分布の標準偏差をそれぞれσ、σ'とすると以下の式が得られる。

$$\begin{aligned}\sigma - \sigma' &= (\bar{f^2} - (\bar{f})^2) - (\bar{f^2}' - (\bar{f}')^2) \\ &= (\bar{f^2} - \bar{f^2}') + ((\bar{f}')^2 - (\bar{f})^2) \\ &= (\bar{f^2} - \bar{f^2}') + (\bar{f}' - \bar{f})(\bar{f}' + \bar{f}). \end{aligned} \tag{4}$$

$(\bar{f^2} - \bar{f^2}') = (\bar{f^2} - \bar{f^3}/\bar{f}) \geq 0$ を容易に示すことができるので、$\sigma' \leq \sigma$となり、結局 selection 操作によって遺伝子の適合度の標準偏差の値は単調に減少することになる。また、スキーマの数については selection によって新しいものが生成されないので単調減少することは明らかである。

crossover と mutation については、問題やコーディング方法に強く依存するので一般的な解析は困難である。そこで、適合度分布の標準偏差と含まれるスキーマの数を、実際の問題について観測した実験結果について、その典型的な例を紹介し、この仮説の妥当性について検討する。図 1 は $f(x) = x$ をある定義域で最大化する単純な最適化問題を GA によって解いた場合の各世代毎の標準偏差の値とスキーマの数の推移を示したものである。

この場合の実験条件は、遺伝子数(個体数)が 50、各遺伝子の長さは 10 であり、アルファベットとして {0,1} の 2 つの文字を使用している。コーディングは遺伝子をあらわす文字列を符合なし 2 進数と解釈するものである。GA のパラメータは

それぞれ $p_c = 0.6$(crossover を行なう遺伝子の割合)、$p_m = 0.01$(mutation を行なう確率) である。図の横軸は GA の世代数を表しており、縦軸はそれぞれ適合度の標準偏差の値とスキーマの数を表している。

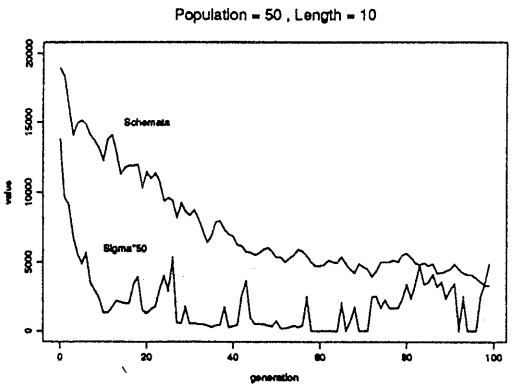

図 1: $\max f(x) = x$ を解いた場合の結果

図 2 は、$f(x) = \sin(40\pi x) + 1$ という多峰性の関数を最適化した場合の結果である。他の実験条件は図 1 と同じである。

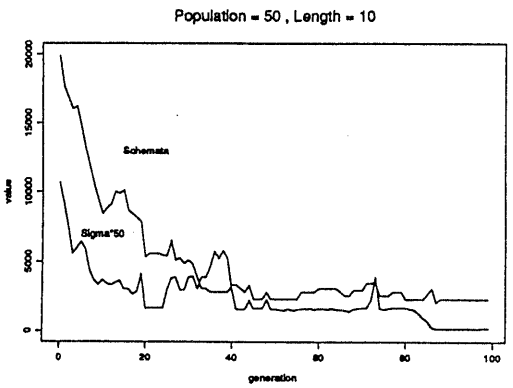

図 2: $\max f(x) = \sin(40\pi x) + 1$ を解いた場合の結果

図 1、2 から分かるように、始めのうちは、スキーマの数の減少と遺伝子集団の適合度分布の標準偏差が、ともに減少している。ある程度以上収束した後はスキーマの数がほとんど変化しなくなるのに対して、標準偏差の値については時々大きな値になってすぐに減少していく過渡的な様子が見られる。これは、crossover や mutation によって新しい遺伝子が生成されたが、適合度が小さいためにすぐに消滅していることを示している。

以上の実験結果に代表されるように、実験によって Sigma-Hypothesis の妥当性を支持する結果が得られた。さらに大きな遺伝子集団についての実験も試みたが、スキーマの数を計るために非常に多くのメモリーを消費するため、実験を行なうことができなかった。また、この結果を得るのに Sun SPARCstation2 でそれぞれほぼ丸一日の計算時間を要している。

5 Sigma-Exchange

我々の提案する効率的な遺伝子交換アルゴリズムである *Sigma-Exchange* についてその詳細を述べる。このアルゴリズムは Random-Exchange と同様に、遺伝子集団をいくつかの部分集団に分割し、そのそれぞれにおいて並列に GA を実行する。Random-Exchange との相違点は、遺伝子交換を起動する条件にある。

上で述べた Sigma-Hypothesis より、標準偏差の値が減少している時、部分集団内に含まれるスキーマの数も減少していると考えられるので、GA のスキーマ処理効率が全体として低下しつつあることが推測できる。従って、各部分集団で適合度分布の標準偏差を常時観測し、その値がある一定割合減少した時にはじめて他の部分集団と遺伝子を交換し、遺伝子の多様性を回復する。

これまでの Random-Exchange では全くランダムに起動していた交換を、交換が必要な時、すなわちスキーマ処理効率が実際に低下した時にのみに起動することで、不必要な交換を防いでいる。実際の並列計算機へのインプリメントを考えた場合、プロセッサ間の通信をできるだけ少なくすることは並列処理の効率を高める上で本質的である。

我々のアルゴリズムを具体的に示したものが Appendix-B である。まず、遺伝子集団を初期化し (Appendix-B の 2 行目)、それぞれ分割された部分集団毎に標準偏差の初期値 $Sigma0$ を計算する

(3~5 行目)。その後、Random-Exchange と同様に各部分集団毎に GA を並列に実行する (8~10 行目)。現在の適合度分布の標準偏差の値 $Sigma$ を各部分集団毎に計算し (11 行目)、その値が初期値に比べてある一定割合 $Lambda$ ($0 < Lambda < 1$) 未満になっていた場合に遺伝子交換を起動する (12 行目)。交換相手となる部分集団をランダムに選び交換を行なう (13~14 行目)。最後に、標準偏差の初期値を再計算する (15 行目)。これらの操作を通常の GA と同様に、ある終了条件が満たされるまで繰り返す (18 行目)。

6 評価実験の結果

我々はこれまで DeJong のテスト関数 F1~F5[2] による関数最適化問題、巡回セールスマン問題 (TSP)、ナップサック問題などについてこの遺伝子交換アルゴリズムの有効性を支持する結果を得ている。

ここでは TSP とナップサック問題に関してその典型的な実験結果を示すこととする。

図 3 に 24 都市の TSP を解いた例を示す。遺伝子集団として 1000 個の遺伝子を用い、それを 20 個のグループに分割した。従って各部分集団にはそれぞれ 50 個の遺伝子が含まれていることになる。コーディングは Grefenstette のコーディング [4] を用いている。また、GA の諸パラメータは、$p_c = 0.6$(Crossover を行なう遺伝子の割合)、$p_m = 0.01$(Mutation を行なう確率)、$p_e = 0.2$(1 回の遺伝子交換で交換される遺伝子の割合) である。

図の横軸は 100 世代終了までに起動された遺伝子交換の回数で、縦軸はその時点で得られた遺伝子集団の中で最良の解の値である。同じ交換回数でどれだけ良い解が得られるかを見るために、Random-Exchange においては交換を行なう確率、Sigma-Exchange においては $Lambda$ パラメータを変化させることで交換回数を調節し、データを取得した。TSP は最小化問題なので値が小さいほど良い解が得られていることになる。この問題の場合、最適解は 820.624 である。この実験で得られた結果によると、Sigma-Exchange を用いることで、単純な Random-Exchange に比べて 10–20%良い解が同じ遺伝子交換回数で得られていることが分かる。

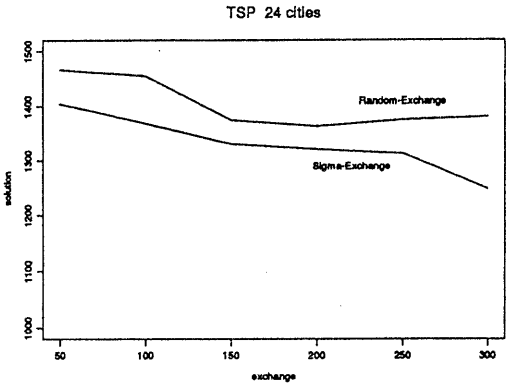

図 3: 24 都市 TSP を解いた場合の結果

また、ナップサック問題についても同様な実験を行なった。まず、L をナップサックの大きさ、$w_i > 0$ $(1 \leq i \leq n)$ をそれぞれ詰め込む物の大きさとし、$x_i \in \{0, 1\}$ $(1 \leq i \leq n)$ を i 番目の物を詰め込む時 1、詰め込まない時 0 として定義する。ナップサック問題は以下の関数 f の最大化問題として定式化できる。

$$f = \begin{cases} 0 & (\sum_{i=1}^{n} w_i x_i > L) \\ \sum_{i=1}^{n} w_i x_i & \text{otherwise} \end{cases} \quad (5)$$

ここでは、$L = 100$、w_i を $w_i \in [5, 25)(1 \leq i \leq 30)$ でランダムに決定した場合の結果を図 4 に示す。コーディングは x_i の列をそのまま遺伝子としたものである。従って、遺伝子の長さは 30 となる。その他の実験条件は TSP の場合と同じである。

この場合の最適解は 100 である。ナップサック問題は最大化問題であるので、大きい値がより最適解に近い。この結果を見ると、Sigma-Exchange が Random-Exchange に比べて 10%程度良い解を、同じ交換回数で得ていることが分かる。

7 まとめ

本研究では、遺伝子集団の適合度の標準偏差とその中に含まれるスキーマの数の関係に関する仮説である Sigma-Hypothesis を提案し、実験によってその妥当性を確かめるとともに、selection に関し

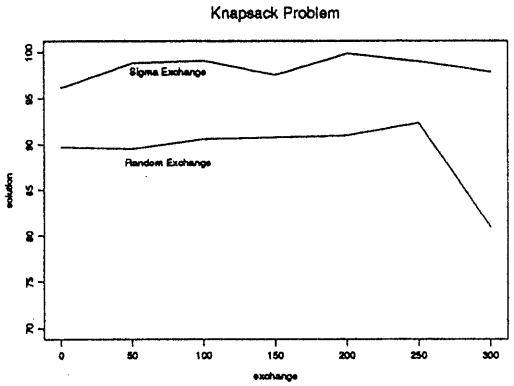

図 4: ナップサック問題を解いた場合の結果

て理論的な考察を行なった。さらに、この仮説に基づいた遺伝子交換アルゴリズム Sigma-Exchange を提案し、遺伝子交換アルゴリズムの改良を行なった。いくつかの問題についての実験を通して、我々の提案したアルゴリズムが有効な手段であることが確かめられた。

今後の課題としては、まず交換相手となる集団の選択と交換の対象となる遺伝子の選択方法に関する改良があげられる。例えば、遺伝子を交換する際に良い遺伝子をコピーして交換する方法、部分集団内の適合度分布の標準偏差の小さいものと大きいものを交換する方法などが考えられる。

また、これらの実験で得られた結果を実際の並列計算機の設計に生かし、並列 GA に適した並列計算機を実現することが本研究の最終的な目標である。適合度の計算はそれぞれ解くべき問題によって異なる計算をしなくてはならなく、GA 自体の操作に比べて複雑な計算を必要とする場合が多い。従って、粒度の非常に小さい超並列計算機は GA の並列化に適さない。そこで、比較的能力の大きいプロセッサを数十個程度用いた並列計算機が適しているものと考えられる。そのような計算機に適合するアルゴリズムとして今回述べたような遺伝子集団分割型の並列 GA がある。その場合にどの程度の遺伝子交換、すなわちプロセッサ間通信や同期が必要であるかについて調べることが重要である。これについて今後さらに研究を進める予定である。

参考文献

[1] COLLINS, R. J. and JEFFERSON, D. R. Selection in Massively Parallel Genetic Algorithms, Proceedings of the Fourth International Conference on Genetic Algorithms (ed.Belew, R. K.), Morgan Kaufmann Publishers (1991), 244–248.

[2] DEJONG, K. A. *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*, PhD thesis, University of Michigan (1975).

[3] GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley (1989).

[4] GREFENSTETTE, J. and GOPAL, R. Genetic Algorithms for the Traveling Salesman Problem, Proceedings of the First International Conference on Genetic Algorithms (ed.Grefenstette, J. J.), Lawrence Erlbaum Associations, Publishers (1985), 160–168.

[5] HOLLAND, J. H. *Adaptation in Natural and Artificial Systems*, University of Michigan Press (1975).

[6] MANDERICK, B. and SPIESSENS, P. Fine-Grained Parallel Genetic Algorithms, Proceedings of the Third International Conference on Genetic Algorithms (ed.Schaffer, J. D.), Morgan Kaufmann Publishers (1989), 428–433.

[7] PETTEY, C. C. and LEUZE, M. R. Theoretical Investigation of A Parallel Genetic Algorithm, Proceedings of the Third International Conference on Genetic Algorithms (ed.Schaffer, J. D.), Morgan Kaufmann Publishers (1989), 398–405.

[8] TANESE, R. Parallel Genetic Algorithm for a Hypercube, Proceedings of the Second International Conference on Genetic Algorithms (ed.Grefenstette, J. J.) (1987), 177–183.

Appendix-A

[Parallel Genetic Algorithm with Random Exchange]

```
1   begin
2   Initialize;
3   do
4       parallel_for i=1 to Number_of_subpopulations do
5           Selection(i);
6           Crossover(i);
7           Mutation(i);
8           if  Random[0,1] < Prob_of_exchange  do
9               j := Random(1..Number_of_subpopulations);
10              Exchange(i,j);
11          endif;
12      endfor;
13  while  Terminate_condition != TRUE;
14  end.
```

Appendix-B

[Parallel Genetic Algorithm with Sigma Exchange]

```
1   begin
2   Initialize;
3   parallel_for i=1 to Number_of_subpopulations do
4       Calculate Sigma0(i);
5   endfor;
6   do
7       parallel_for i=1 to Number_of_subpopulations do
8           Selection(i);
9           Crossover(i);
10          Mutation(i);
11          Calculate Sigma(i);
12          if  Sigma(i) < Sigma0(i)*Lambda  do
13              j := Random(1..Number_of_subpopulations);
14              Exchange(i,j);
15              Calculate Sigma0(i) and Sigma0(j);
16          endif
17      endfor;
18  while  Terminate_condition != TRUE;
19  end.
```