

**A Study on Parallel Distributed Genetic Algorithms
(Discussion on a Randomized Migration Island Model of Distributed Genetic Algorithms)**

Mitsunori MIKI^{*3}, Tomoyuki HIROYASU^{*3}, and Yasunori NAKAMURA^{*3}

^{*3}*Doshisha University, Dept. of Knowledge Engineering, Kyo-Tanabe, Kyoto, 610-0321, Japan*

This paper discusses about the characteristics of distributed genetic algorithms (DGAs). Among the several types of models for DGAs, this paper focused on a randomized migration island model. In this model, the island of the migration is decided as every migration opportunity at random. When there are a lot of islands, this model is very useful. The efficiency of this model is discussed through the numerical examples. In this study, Distributed Genetic Algorithm with Distributed Environment (DEGA) is also introduced. Usually, the parameters in GAs are the same in each island but they are different in DEGAs. This approach makes designers free from setting appropriate GA parameters. Applying this algorithm to solve numerical examples, it is also clarified that there are several advantages in this approach.

Key Words : Genetic Algorithms, Distributed Processing, Parallel Computing, Island model, Optimum Design

1 . 緒言

遺伝的アルゴリズム(Genetic Algorithm: GA)は生物の進化のしくみを模倣した確率的多点探索手法である。GAは設計空間が離散空間の場合にも対応できる探索であるために多峰性にも一般的には強いとされている。

一方でGAは高品質な解を探索するためには、数多の個体を用意し多くの世代にわたって繰り返し計算を繰り返すため計算負荷が高くなり計算時間の短縮が重要な課題となる。これに対する解決法の一つが並列処理によりGAの計算時間の短縮を図る方法である。GAの並列化の方法はいくつか考えられるが、代表的なものの一つとしてGAの母集団をいくつかのサブ母集団(島)に分割して適度な世代間隔(移住間隔)でサブ母集団ごとにサブ母集団の個体数に移住率を決定される複数個の遺伝子の交換(移住)を行ういわゆる分散遺伝的アルゴリズム(Distributed Genetic Algorithm: DGA⁽²⁾)を並列に処理する手法が挙げられる。この手法は遺伝子の交換が頻繁に行われるわけではないので、通信によるオーバーヘッドはそれほど

問題にならず、並列化効率も上がりGAの特徴を活かせるという利点がある。

こうした計算時間の短縮という利点とは別にDGAでは、個体進化における早熟を避け、世代が進んでも個体の多様性を維持することができ、得られる最終解の高品質化が達成できるという利点が報告されている⁽³⁾。これに対して、三木らは工学的な問題に対してDGAの基本的な特性を明らかにする目的で、構造最適化の分野における問題を対象に、サブ母集団の個体数、移住頻度などの影響を検討した⁽⁴⁾。その結果、問題のクラスによっては、適切な移住率および移住間隔を選択する事によりDGAは単一母集団のGAより高い品質の最適解をもたらす場合があること、高品質化は分割された母集団での進化によりそれぞれ最適化した局所最適解が探索されそれが移住により組み合わせられることで大局的最適解になることなどが明らかになった。

三木らが使用した分散モデルは踏み石型モデルであり、各サブ母集団はリング状に連結されている。隣りの島へ順次一方向へ移住する方式としている。すなわち、各移住においては島の個体数に移住率

非常に多い計算環境,すなわち超並列計算環境
計算を行うことも多くなっており,そのような場
は島数を増やすことでさらに多様性が保持できる
さらなる広域探索が可能であると考えられる.
しながら,島が多くなると踏み石型モデルでは,
に対して個体の移住がほとんど影響しなくなるも
考えられる.そこで本報ではこれまでの隣接する
みへの移住を行うのではなくランダムに移住を行
る戦略を採用する.トラス構造設計を対象とした
計算例によりこの効果を検討し考察を行う.さら
れまでは交叉率や突然変異率などといったGAに
な変数の値をすべての島で統一して設定を行って
が,本報では島ごとにそれらのパラメータ値を異
たものを設定する戦略も採用する.これまでパラ
タは解の精度に影響を及ぼしていたため,設計者
ラメータを最適な値に設定することが必要であっ
それに対してこの戦略では,各島で適度に異なる
メータの値を設定しておくことで良好な解が得ら
ことが期待できる.この戦略を採用したものを環
散型遺伝的アルゴリズムと呼ぶ.この戦略の効果
いても数値計算例を通じて検討を行う.

2. 分散遺伝的アルゴリズム

2.1 分散遺伝的アルゴリズム 分散遺伝的アル
ゴリズム(Distributed Genetic Algorithm: DGA)は通常
の母集団を複数のサブ母集団に分割し,このサブ
集団を島と呼ぶ.各島では通常のGAを行い定期
いくつかの個体を島の中から選択し,他の島との
を行う.この操作は移住と呼ばれる.通常移住は
はじめ決定されている世代間隔で行われこの間隔
注間隔と呼ばれる.移住する個体の数は島内の個
に移住率という変数を乗して決定される.これら
注間隔および移住率は設計者の決定するパラメー
ある.アルゴリズムの概要を図1に示す.

注という操作には様々な方式が存在する. Nang
移住先が決定している方式を踏み石型モデル,移
を移住ごとに変更する方式を島モデルと呼んでい
しかしながら,一般には,踏み石型モデルを島
ルと呼んでいる研究も数多く見られる⁽⁶⁾. 本研究
これらを明確に区別するために移住先が決定して
方式を踏み石型モデル,移住先を移住機会ごとに

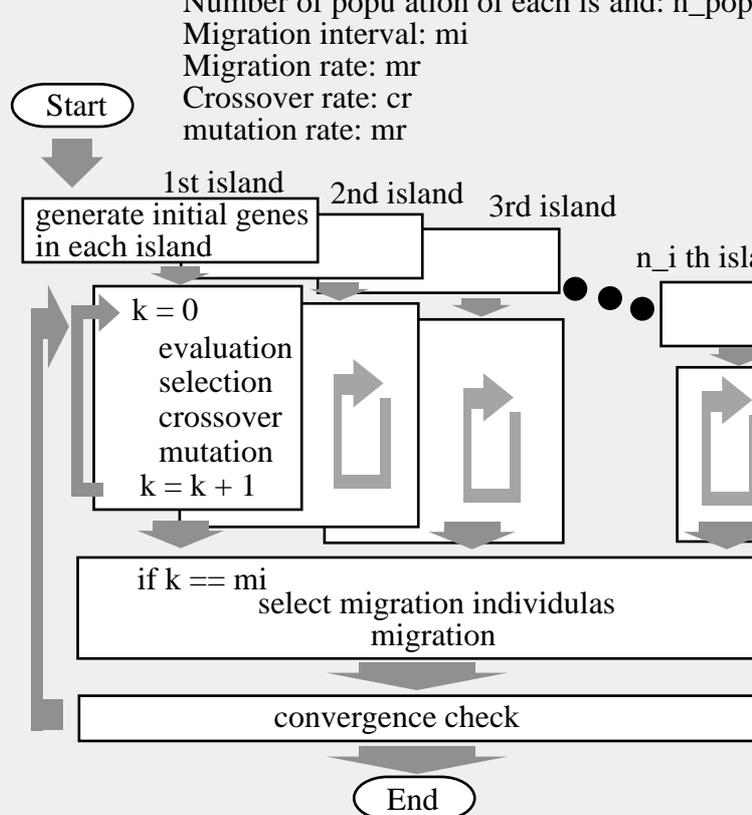


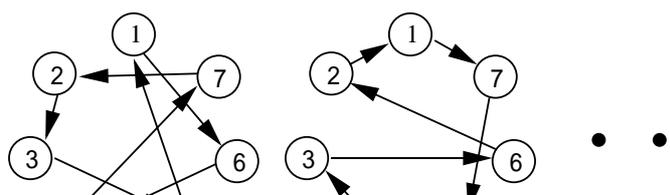
Fig. 1 Flow of distributed genetic algorithm

移住する個体の選択にも様々な方法が存在する
内の最良解のみを移住させる方法⁽⁷⁾, 移住島間で
ナメントを行い勝者が両島でコピーされる方式⁽⁸⁾
良解と移住先の最悪解とを交換する方法⁽⁹⁾などで
本研究では次節で説明する通り,島内から移住に
く個体をランダムに選択し移住させる方式をとる

2.2 ランダム移住型モデル 三木らが報
た踏み石型島モデル⁽⁴⁾では,個体の移住が隣接す
のみで行われるため,ある島内に優位な個体が存
たとしてもその個体が全体に移住して行く可能性
常に低くなり,最適解を得ることが困難になるこ
予想される.

それに対して,本研究で行う移住では,移住間
に各島が異なった島に移住するモデル,ランダム
住モデルを採用する.ここでは全体の島をランダム
順序付け,全体として各島に同時に複数の島から
住は生じないようにしている.これより,移住パ
ンの一例は図2のようになる.

移住においては島の個体数に移住率を乗じた数
体をランダムに選択し,一定の世代交代後に同
とって移住させる方式としている.すなわち,移



世代に残す。すなわち、島内での各個体の適合度計算し、もっとも適合度が低い個体を前世代のエリート個体と入れ換える。その後、同じ数の個体群を島から受け取る。その後、交叉、突然変異、およびエリート方式による淘汰を行う。

3 環境分散型遺伝的アルゴリズム GAに最適解探索においては、広域探索による大域的最適解の探索が可能であるという利点がある一方で、設計者が決定しなければならない問題に依存した変数が存在するという問題が存在する。さらに、各島においては、島の個数、各島における個体数などといった設計者が決定しなければならない変数がさら増加することとなる。それに対して、本研究では、突然変異率や突然変異率といった変数の値を各島によって調整させて設定するアプローチをとる。

ここで、各島のパラメータの値を異なるものに設定することで、いずれかの島の値は最適値に近いものになることが期待でき、そこで得られた個体は移住によって各島へと伝播する。そのため、これまで設計者が調整して1つの値に設定していた変数の値は細かな調整をすることなく設定することが可能となり、設計者の負担を軽減することが可能であると考えられる。本研究では環境分散型遺伝的アルゴリズム(Distributed Genetic Algorithm with Distributed Environment: DEGA)とすることとする。このスキームを選択することによって各島は異なった特性と環境を持つこととなり、各島で異なった局所解の探索が行われる可能性が高くなる。さらに、島ごとに移住をくり返すことにより、高い適合度の最適解が探索可能となることが予想される。

3. 数値実験

本章で提案したランダム移住型島モデルとDEGAの性能を検討するために次に示すような2次元トラス構造の重量最小化設計問題を通じて検討する。

3.1 構造最適設計問題 計算の対象とした問題は図3に示す11部材の最小体積問題とした。目的関数は各部材の体積の合計であり、制約条件は各部材の引張応力および座屈応力、ならびに節点6における変位制約とした。これと同種の問題は多いが、ほとんどは部材の引張と圧縮応力制約のみで、座屈応力や変位制約を考慮しているものは少ない。変位制

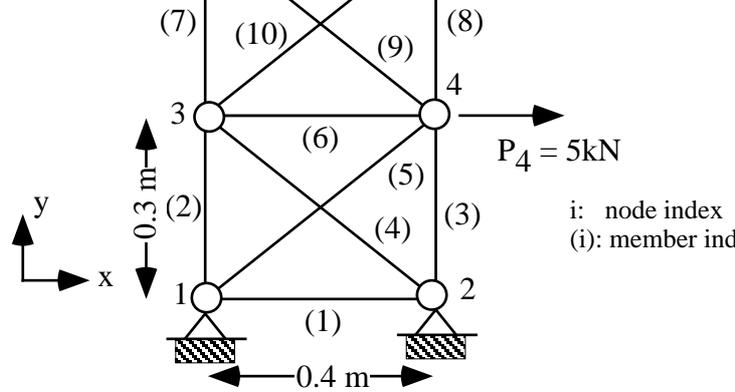


Fig. 3 11member truss

設計変数は円形部材の断面積とし、変域を10000mm²とし、これを12ビットで表現する。最小すべき適合度関数は次式に示す。

$$H = \frac{1}{W_v V_T + P_d + P_t + P_b} \dots\dots\dots (1)$$

ここで

$$P_d = W_d \times d_6^2 \quad \text{if } (d_6 > 0.03 [m]) \dots\dots\dots (2)$$

$$P_t = \sum_{i=1}^{\text{number of design variables}} P_{ti} \quad P_{ti} = 1 \quad \text{if } (\sigma_i > 40 [MPa])$$

$$P_{ti} = 0 \quad \text{otherwise} \dots\dots\dots (3)$$

$$P_b = \sum_{i=1}^{\text{number of design variables}} P_{bi} \quad P_{bi} = 1 \quad \text{if } (L_i > L_c)$$

$$P_{bi} = 0 \quad \text{otherwise} \dots\dots\dots (4)$$

V_T は部材体積の合計、 W_v は重み係数(=60)、 P はそれぞれの変位(d)、引張(t)、および座屈(b)の制約条件とするペナルティ関数である。変位に関するペナルティは変位の2乗に重み係数 W_d (=1000) をかけのとし、引張応力および座屈に関するペナルティは制約条件が満足されないときに一定値を足す方向とした。この方法を用いない場合には局所制約を満足させる解を見いだすことは難しくなる。

また、本研究では1対象問題のみを取り扱ってが、ここで得られる結果は、同規模のトラス構造体積問題のようなクラスの問題においてあてはまのと考えられる。

使用した並列計算機は64プロセッサを搭載した国nCUBE社製のnCUBE2Eである。プログラムを実行するにあたり、使用した言語はプロセッサ間通信のための関数が追加されたnCUBE C++である。1試行あたり、1CPUの場合、10試行平均 1.118×10^5 [s]を要した。

3.2 ランダム移住型 DGA の効果 ランダム移住型 DGA の効果を検討するために前節で示した2次元トラス最小体積問題に適用し、解を求め

交叉のGAとし、予備的な検討により交叉率0.6、
変異率0.01とした。世代数が100を越えた時点で
する。また、以下に示す各結果は10回の試行の平均
である。

に踏み石型モデルの最良解を図5にランダム移
モデルの最良解を移住間隔、移住率、島数を変化
した時の結果を示す。図4、図5からわかるように、
的傾向として、移住間隔が30よりも10の方が
な解が得られている。また、踏み石型モデルに対
ランダム移住型モデルの方が良好な解が得られて
といえる。ここで、両者のモデルで得られた解の比
明確にするため、ランダム移住型モデルで得られ
の適合度を踏み石型モデルで得られたもので除し
を島数を変化させて図6に示す。この図ではラン
型モデルで得られた適合度が踏み石型モデルで得
たものよりも良好な際に1.0以上となる。

からわかるようにほとんどの移住パラメータで
が増加するにつれランダム移住型モデルが踏み石
モデルを上回る傾向にあり、特に移住間隔0.3、移住
が大きく踏み石型モデルを上回っている。また、

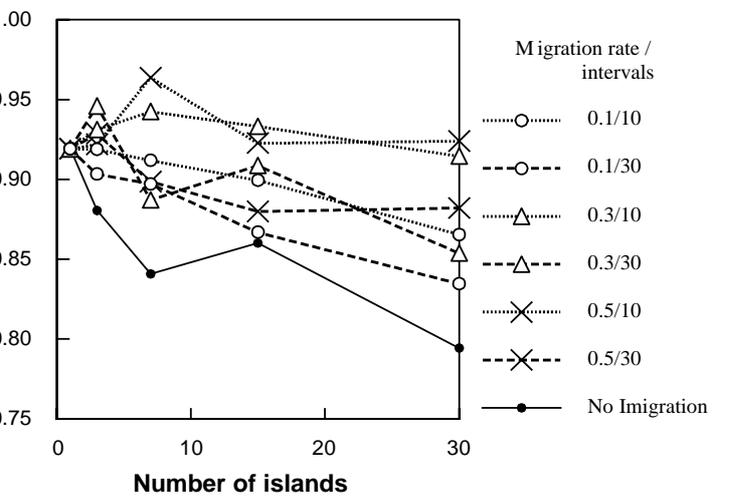


Fig. 4 Maximum fitness value of stepping stone model

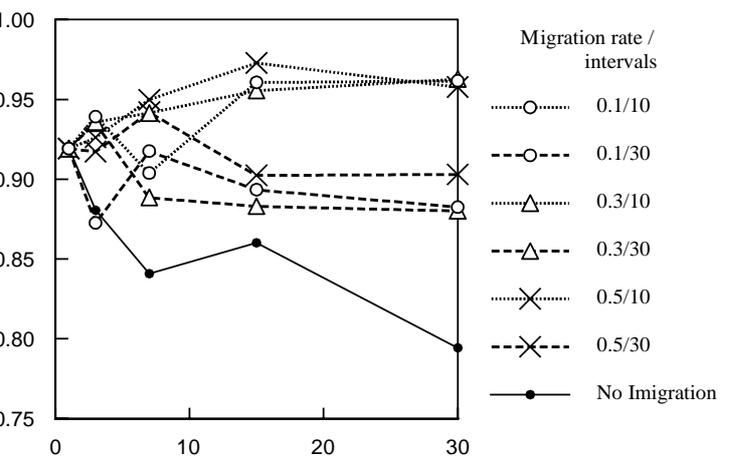


Fig. 5 Maximum fitness value of random migration model

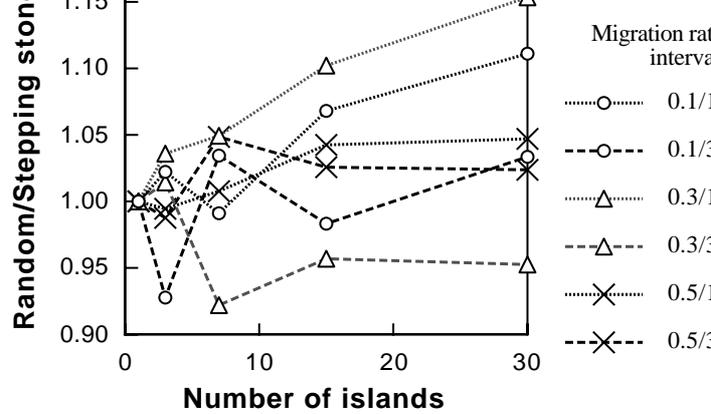


Fig. 6 Comparing of fitness values of Random and Stepping stone

島数の増加につれ値が大きく低下するものがない
より、ランダム移住型モデルは踏み石型モデルよ
島数の多い、すなわち、より分散化したモデルに
ているといえる。これは先に予想したように島が
なるとこれまでの方法では、全体に対して個体の
がほとんど影響しなくなるからであると考えられ
これを確認するために、比較的島数の多いと考
れる30島の場合の各島で100世代後に得られる最
体の設計変数の様子を踏み石型モデルの場合とラ
ム移住型モデルとの場合を図7および図8に示す

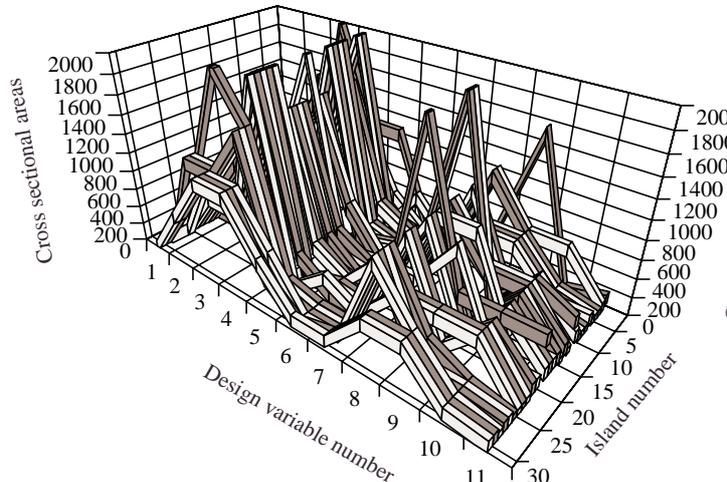
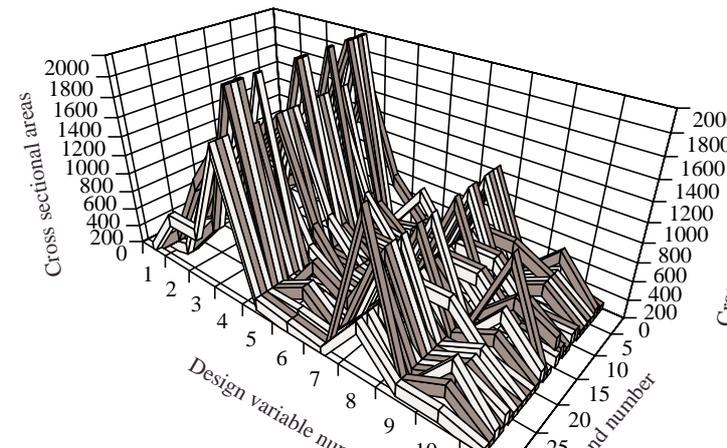


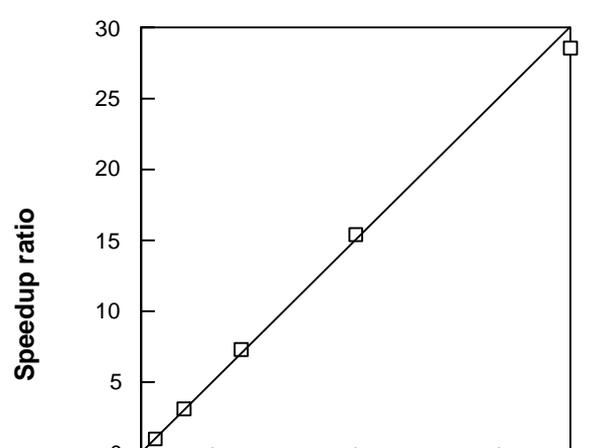
Fig. 7 Details of good solutions in real design variable space (stepping stone model)



個体が均一でなく、それに対してランダム移住型モデルでは各島で得られる最良個体がほぼ均一となる。これは島数が多い場合に予想した通り踏み石モデルでは移住による個体情報が全島に行き渡っているのに対して、ランダム移住型モデルにおいて個体情報が全島に行き渡っていることを示している。全島の最良個体が均一の際には初期収束の可能性があるが、踏み石型モデルとランダム移住型モデルとで得られた解の適合度を図6などより比較するとランダム移住型モデルによって得られた適合度の方が良いことから、初期収束をしている状態ではないことがわ

る。この実験において島数は一概には多い場合に良好な結果が得られているとは言えない。例えば、移住率0.5に設定して、移住間隔10の場合には、島数が15の時に最良解が得られている。これにより、パラメータおよび島数によって最適な島数が存在するものと考えられる。最適な島数がどのようなパラメータに依存するかの検討は今後の課題である。

ここでは本研究で得られた島数と計算時間の関係を示している。横軸に島数を縦軸には速度比を示す。速度比とはプロセッサ数が1の際の計算消費時間を各使用プロセッサ数での計算消費時間を除した値であり、本研究では島数とプロセッサ数が同一であるので、島数と速度比の関係は線形が理想的である。図9には理想的なスピードアップが実線として記してある。この図からも分かる通りほぼ線形的なスピードアップが実現されている。特に15島まではスピードアップが島数にほぼ等しいような非常に良好な結果となった。このように、島モデルによる並列化が通信量が少ないというだけでなく、分散化することで、収束性そのものが向上するという性質によるものと考えられる。



(DEGA)の効果を検討するために、前節と同様に30島で示したトラス構造問題に適応し検討する。

実際の問題に適応する際には、交叉率および突然変異率のそれぞれを各島ごとに変化させるものとなるが、本数値実験においては、それぞれの効果を検証するために、交叉率一定の下に突然変異率を各島で変化させる場合と突然変異率一定の下で交叉率を各島で変化させる場合を行う。また、各島に設定する値は適当な範囲から適当に選択したものを設定することになっている。

ここでは、島数30で行い、突然変異率一定で交叉率が異なるDEGAの結果を図10に、交叉率一定で突然変異率が異なるDEGAの結果を図11に示した。

突然変異率一定で交叉率が異なるDEGAでは、10ごとにそれぞれ交叉率として0.3, 0.6, 0.8を割って示した。図10には島ごとに移住率を変化させた場合(Variable DEGAと表記)、移住率が一定のもの(それぞれ0.3, 0.6, 0.8と表記)、移住率が変動するもの(Variable DEGAと表記)の結果とともに示している。変動型では、

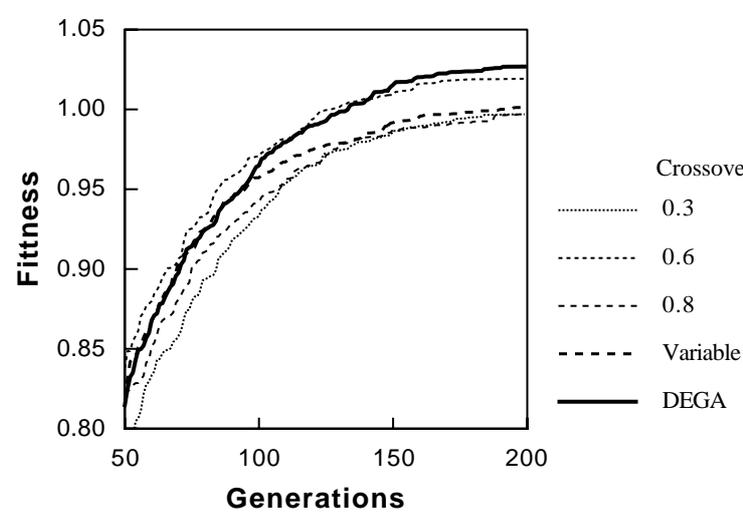
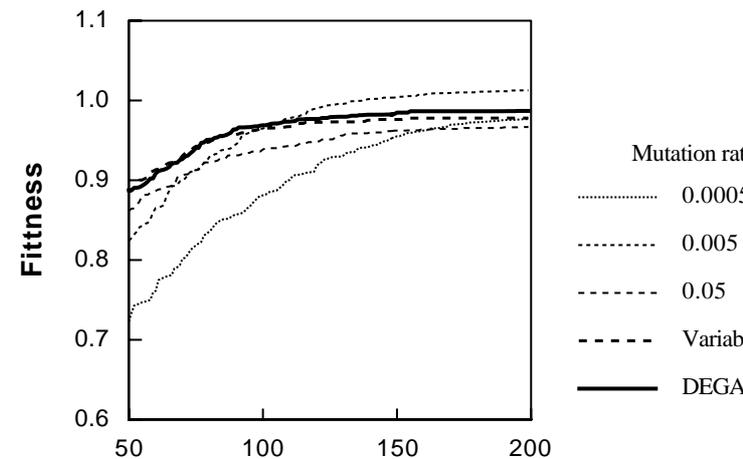


Fig. 10 Performance of distributed crossover rate scheme



る。

交叉率一定で突然変異率が異なる DEGA では島数ごとにそれぞれ突然変異率として 0.05, 0.005, 0.0005 を割り当てた。図 11 には 10 島ごとに突然変異率をそれぞれ 0.05, 0.005, 0.0005 (それぞれ 0.05, 0.005 と表記), および変動型 (Variable と表記) のものとともにその適合度関数値の世代ごとの変化を記してある。変動型では, 初期 50 世代では 0.05 から 0.05 へ次 50 世代では 0.005 から 0.0005 へ変化させ以降の世代では収束するまで一定に減少させている。10 から DEGA によって得られた結果は交叉率一定の変動型のものよりも良好な結果となっている。GA において交叉率などのパラメータの設定は大きな問題となる。ここで得られた結果は, DEGA により設計者の交叉率の設定は適当に分散させて設定することが必要で, 最適な値を設定することは不必要であることを示している。

一方で, 突然変異率を変化させた場合には, DEGA によって必ずしも最適解が得られる場合ばかりでないことが図 11 より明かである。しかしながら, 突然変異率を適当に設定されていない場合や変動型では, 局所最適解が探索されてしまう可能性がある。通常, 最適解の探索は突然変異率の問題に依存するために, 解の探索前に最適な値を見つけることは非常に難しい。それに対し DEGA では, 突然変異率をあらかじめ最適な値と決定する必要がなく, 結果的にある程度の解が探索できるので非常に有効な手法であると言えよう。

4 . 結言

本研究では, 踏み石型モデルによる分散遺伝的アルゴリズムに対して, ランダム移住型分散遺伝的アルゴリズムによるトラス構造最小体積問題を対象として検討を行った。その結果, トラス構造最小体積問題のよみ石型モデルと比較して良好な解が得られる。島数が多い場合には踏み石型モデルでは移住によ

Algorithm with Distributed Environment: DEGA) を提案した。この手法に対してもトラス構造体積最小化問題に適用することで以下の事項が明かとなった。

- 3) DEGA は最適な交叉率を設定して得られる結果はほぼ等しい解が得られ非常に有効であるといえる。
- 4) DEGA により最適な突然変異率を設定して得られる解が得られるわけではない。しかし, その突然変異率を設定して得られる解よりも良好な解が得られ, 突然変異率を最適な値に設定するの点と比較すると DEGA によって得られる解は良好な解であると言える。
- 5) DEGA はこれらの結果から, これまで GA の最適点とされていた交叉率や突然変異率などを, 設計者が設定する必要のない有効な手法であると言える。

参考文献

- (1) Goldberg, D. E., Genetic algorithms in search, optimization, and machine learning, Addison-Wesley, (1989).
- (2) Tanase, R., "Distributed Genetic Algorithms," 3rd Int. Conf. Genetic Algorithms, (1989), 434.
- (3) Belding, T.C., "The Distributed Genetic Algorithm," Proc. 6th Int. Conf. Genetic Algorithms (1995), 114.
- (4) 三木・他 2 名, 機論, 65-638, A(1999), 217.
- (5) Nang, J. and Matsuo, K., "A Survey on the Parallel Genetic Algorithms," J. SICE, Vol. 33, No.6, (1994).
- (6) Whitley, D., et. al., "Island Model Genetic Algorithms and Linearly Separable Problems," Proc. of AISB Workshop on Evolutionary Computation, (1997).
- (7) Marin, F. J., et. al., "Genetic Algorithms on LAN-Message Passing Architectures using PVM: Application to the Routing Problem," Proc. of Parallel Problem Solving from Nature, (1994), 534.
- (8) Gordon, V. S. and Whiteley, D., "Serial and Parallel Genetic Algorithms as Function Optimizers," Proc. Int. Conf. Genetic Algorithms, (1993).
- (9) Starkweather, T., et. al., "Optimization using distributed genetic algorithms," Proc. of Parallel Problem Solving from Nature, (1991), 176.
- (10) Miki, M., "Object-Oriented Approach to Mod

pp. 972-977