JAIST Repository
https://dspace.jaist.ac.jp/

Title	並列 GA における多様な部分集団群による協調探索の効果
Author(s)	堀井, 宏祐
Citation	
Issue Date	1998-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1124
Rights	
Description	國藤進, 情報科学研究科, 修士

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

修 士 論 文

並列 GA における
多様な部分集団群による協調探索の効果

指導教官　國藤進 教授

北陸先端科学技術大学院大学
情報科学研究科情報処理学専攻

堀井宏祐

1998 年 2 月 13 日

Copyright © 1998 by Hirosuke HORII

目 次

1 はじめに **1**

2 遺伝的アルゴリズムとその並列化の概要 **3**

2.1 遺伝的アルゴリズムの概要 3

2.1.1 選択 4

2.1.2 交叉 6

2.1.3 突然変移 7

2.2 並列遺伝的アルゴリズム 7

2.2.1 並列 GA の目的 7

2.2.2 並列 GA の分類 8

3 非同期並列遺伝的アルゴリズム **12**

3.1 移住操作 12

3.1.1 探索状況の把握 13

3.1.2 バイアス，一時スキーマ 13

3.1.3 本研究における移住操作の枠組 15

3.2 多様な部分集団群による協調探索 15

4 実験 1 : ナップザック問題への適用 **16**

4.1 ナップザック問題 16

4.2 実験方法 16

4.3 実験結果および考察 17

5 実験 2 : ロイヤル・ロード 関数への適用 **23**

5.1 ロイヤル・ロード関数 23
5.2 実験 2.1 : 移住方法の検証 26
5.2.1 実験方法 26
5.2.2 実験結果および考察 26
5.3 実験 2.2 : 異質な部分集団群による探索の検証 27
5.3.1 実験方法 27
5.3.2 実験結果および考察 29

6 おわりに **31**

謝辞 **32**

参考文献 **33**

第 1 章

はじめに

遺伝的アルゴリズムは自然界のシステムの適応過程を説明し, 進化のメカニズムを模擬する人工モデルとして, J. H. Holland によって考案されたアルゴリズムである. 現在では進化のモデルとしてだけでなく, 確率的探索, 学習, 最適化の一手法として, 制御, スケジューリング, 最適設計のような工学的応用の分野においても様々な研究がなされている [Goldberg 89].

遺伝的アルゴリズムは, 選択, 交叉, 突然変移といった遺伝的操作, 各個体の適応度計算を何世代にもわたっておこなうため, 膨大な計算量が必要とされる. しかし, 各個体は同時並行に遺伝的操作, 適応度計算を実行することが可能であり, 遺伝的アルゴリズムは並列性の高いアルゴリズムであるといえる. そのため, 研究の初期の頃から様々な並列化手法が提案されており, その手法は大域並列化, 島モデル並列化, 大規模並列化の 3 種に分類される.

島モデル並列遺伝的アルゴリズムは, 集団を複数の部分集団に分割し, 並列計算機上の各 PE に部分集団を割り当て, 一定期間, 条件により, 移住操作を起動し, 他の部分集団から移住者を受け入れるアルゴリズムである.

また, 遺伝的アルゴリズムの問題点の 1 つとして, 過剰収束があげられる. これは, 適応度の高い個体のもつ遺伝子が急速に集団中に広がることにより, 集団中の遺伝子の均一化がおこり, 進化の停滞を招く問題である. 集団遺伝学における知見として,2 つの集団が地理的に隔離されて遺伝子の交換が妨げられ, 遺伝的浮動によって生殖的な隔離に至る, 地理的種形成というものがある. 島モデル並列遺伝的アルゴリズムにおいても集団間の移住操作を制御することによって, 各部分集団が異なる遺伝的構成をもつようになり, 集団全

体として多様性を維持することができると考えられる.

島モデル並列遺伝的アルゴリズムにおいては, 部分集団間の協調が最も考慮すべき問題である. 一定期間または条件ごとに部分集団間でおこなわれる移住により, 他集団の遺伝子を受け入れることによって, 部分集団中の遺伝子の多様性を維持し過剰収束を回避するが, 移住の間隔, 条件, 移住者の選択方法の設定が探索能力に大きな影響を与える. 各部分集団における探索中に他の部分集団からの移住者が導入された場合, 集団中の遺伝子の多様性は回復するが, 良好なスキーマが破壊される危険性がある. そのため, 移住者の導入は各部分集団内における探索が終了した後におこなうことが効果的である. また, 各部分集団によって探索の進度が異なるのは自明であり, 各部分集団が各々の探索状況に応じて非同期に移住をおこなうことが必要である.

本研究では, 各部分集団が各々の探索状況に応じて非同期に移住操作をおこなう非同期並列 GA (Asynchronously Parallel GA) において, 以下の提案をおこなう.

- 部分集団の遺伝的構成の一様化を移住操作の起動条件とし, 一様化の検出方法としてバイアスを利用する.
- 自分と異なる遺伝的構成をもつ部分集団から移住者を導入し, 部分集団の遺伝的構成を把握する指標として一時スキーマを利用し, 一時スキーマのハミング距離によって, 遺伝的構成の相違度を判定する.
- 各部分集団に異なる遺伝的操作のパラメータを設定することによって, 明示的に異なる探索傾向をもたせる.

以上の提案をふまえた並列遺伝的アルゴリズムを CRAY-T3E 上に実装し，ナップザック問題, ロイヤル・ロード関数に対して適用することにより，その効果を検証する.

第 2 章

遺伝的アルゴリズムとその並列化の概要

2.1 遺伝的アルゴリズムの概要

遺伝的アルゴリズムは自然界のシステムの適応過程を説明し，生物の進化のメカニズムを模擬する人工モデルとして，J. Holland によって 60 年代に考案された．60 年代から 70 年代にかけて Michigan 大学において Holland の研究グループによって研究が進められ，1975 年に著書 *Adaptation in Natural and Artificial Systems* においてその理論的枠組が示されている．

1985 年には国際会議，International Conference on Genetic Algorithms (ICGA) が開催された．その後， ICGA は 2 年ごとに開催されており，現在では GA は確率的探索，学習，最適化の一手法として広く研究がなされている．

GA の基本的な処理の手順を以下に示す．

1. 初期化: 一定数の個体をランダムに生成し，初期集団とする．
2. 選択: 各個体の適応度を計算し，適応度にもとづいた一定の規則によって親となる個体を選択する．
3. 交叉: 選択された親の染色体を交叉させ，次世代の子を生成する．
4. 突然変移: 一定確率によって個体の一部を変化させる．

選択，交叉，突然変移は遺伝的操作 (Genetic Operator) と呼ばれ，選択→交叉→突然変移のサイクルを 1 世代として，何世代にもわたる集団に対する遺伝的操作によって，優

良な個体を生みだす.

遺伝的操作は探索能力に大きな影響を与えるため，現在に至るまで様々な手法が提案されている．ここでは各遺伝的操作の代表的な手法を概説する．

2.1.1 選択

Darwin は 1859 年に著書 *On the Origin of Species by Means of Natural Selection* において，生存に有利な個体が選ばれ，不利な個体は淘汰されるという適者生存の理論を述べた．GA における選択 (Selection) の基本的な考えも同様であり，適応した個体に子孫を残す機会が多く与えられる．

ルーレット選択

ルーレット選択 (Roulette Selection) は適応度比例選択とも呼ばれる．各個体は適応度に比例した確率で親として選択され，自らの遺伝子を次世代に伝えることができる．個体数 N の集団において，適応度が f_i の個体 i が親として選択される確率 $pselect_i$ は，次式で表される．

$$pselect_i = \frac{f_i}{\sum_{j=1}^{N} f_j}$$

期待値選択

S. Wright は 1930 年ごろから小集団における対立遺伝子の相対頻度の変化に，遺伝的浮動 (Randim Genetic Drift) が大きな影響を与えることを明らかにした．また T. Dobzhansky と O. Pavlovsky は 1957 年に，ウスグロショウジョウバエ (Drosophila pseudoobscura) の 5000 個体の大集団と 20 個体の小集団の個体数の異なる集団間で，遺伝子配列の異なる 2 種類の染色体の頻度分布を比較した．その結果，大集団では 10 集団の間でほぼ同じ頻度分布を示したが，小集団では集団によって相対頻度が大きく異なることがわかった．

GA においても確率選択である場合，個体数の小さい集団では適応度よりも乱数の揺らぎに大きく影響を受けた選択がなされる可能性がある．期待値選択 (Expected-Value Selection) はその問題に対処するために提案されたものである．各個体の適応度によって，各個体が残す子孫の数の期待値を計算し，親として選択される度に一定割合で期待値が減少される．これによって，遺伝子の偏りを防止しようとする手法である．

ランキング選択

ルーレット選択や期待値選択では適応度の値が選択確率を定める．しかし，適応度の突出した個体が出現した場合，全ての個体が均一化してしまったり，逆に適応度に差がない場合，進化が停滞してしまう可能性がある．ランキング選択 (Ranking Selection) は適合度によって各個体をランク付けし，順位によって決められた数の個体が選択される．

[Baker 85] では線形ランキング選択が提案されており，個体 i が親として選択される確率 $pselect_i$ は，次式で表される．

$$pselect_i = \frac{1}{N}\left(\eta^+ - (\eta^+ - \eta^-)\cdot\frac{i-1}{N-1}\right)$$

定数 η^+, η^- は最大期待値と最小期待値であり，線形関数の傾きを定める．

[Michalewicz 92] では非線形ランキング選択が提案されており，個体 i が親として選択される確率 $pselect_i$ は，次式で表される．

$$pselect_i = c\cdot(1-c)^{i-1}$$

c はランク 1 の個体が選択される確率である．

トーナメント選択

トーナメント選択 (Tournament Selection) は集団から複数の個体をランダムに選択し，その中で最も適応度の高い個体をトーナメント方式で親として選択する手法である．

エリート保存選択

確率的に選択をおこなう場合，優良な個体が生成されてもすぐに消滅してしまうことがある．エリート保存選択 (Elitist Preserving Selection) は適合度が上位の個体の一定割合を無条件に次世代に残す手法である．エリート保存選択は最良の個体が交叉や突然変移によって破壊されない利点があるが，その半面，エリート個体が急速に集団中に広がる可能性が高く，局所解に陥る危険性がある．そのため，他の手法と組み合わせて適用することが多い．

2.1.2 交叉

交叉 (Crossover) は，親として選択された個体間で染色体の組換えをおこない，新しい個体の生成をおこなう操作である．一般的な GA では各個体が 2 値 $\{0, 1\}$ のビット列で表現されており，交叉は個体間のビット列の交換によっておこなわれる．

1 点交叉

1 点交叉 (One-Point Crossover) は選択された 2 つの親のビット列上の 1 点を交叉点 (Crossover Point) としてランダムに選び，交叉点を境に 2 つの親のビット列を交換することによって，次世代の個体を生成する．1 点交叉の例は以下のようになる．

親 1 : *110110*|*1110*

親 2 : 011001|1011

⇓

子 1 : *110110*|1011

子 2 : 011001|*1110*

多点交叉

多点交叉 (Multi-Point Crossover) が 1 点交叉と異なる点はランダムに交叉点が複数選ばれる点である．交叉点間のビット列を交互に交換し，次世代の個体を生成する．例えば 2 点交叉の場合，以下のようになる．

親 1 : *1101*|*101*|*110*

親 2 : 0110|011|011

⇓

子 1 : *1101*|011|*110*

子 2 : 0110|*101*|011

また，3 点交叉の場合，以下のようになる．

親 1 : *11*|*011*|*0*|*1110*

親 2 : 01|100|1|1011

⇓

子 1 : *11*|100|*0*|1011

子 2 : 01|*011*|1|*1110*

一様交叉

一様交叉 (Uniform Crossover) は [Syswerda 89] において提案された交叉法である．一様交叉では，2 値 $\{0, 1\}$ をランダムに発生させ，遺伝子長と同じ長さのマスクパターンを生成する．そして，マスクパターン上の 1 の遺伝子座にある遺伝子を交換することによって，次世代の個体を生成する．一様交叉の例を以下に示す．

親 1:	*1101101110*
親 2:	0110011011
マスク:	1011011010
子 1:	0*11*0*1*11*1*0
子 2:	*11*0*1*0*0*1*0*1*1

2.1.3 突然変移

選択，交叉のみでは集団中の個体のもつ遺伝子の組合せの範囲のみの探索しかできず，局所解に陥ってしまうと脱出することができない．突然変移 (Mutation) は一定の突然変移率によって遺伝子を変化させ，集団中に含まれない遺伝子を導入し，多様性の維持をはかる手法である．突然変移率が大きすぎると，スキーマが破壊され，ランダムサーチに陥ってしまう．

2.2 並列遺伝的アルゴリズム

2.2.1 並列 GA の目的

GA の並列化には様々な手法があるが，それらに共通する目的として以下の 2 点があげられる．

処理速度の向上

GA は，選択，交叉，突然変移といった遺伝的操作，各個体の適応度計算を何世代にもわたっておこなうため，膨大な計算量が必要とされる．しかし，各個体は同時並行に遺伝的操作，適応度計算を実行することが可能であり，遺伝的アルゴリズムは並列性の高いアルゴリズムであるといえる．そのため，並列計算機上への実装によって処理速度の向上を

はかることが容易である．プロセッサ間の通信コスト，全体の制御法が検討課題としてあげられる．

集団中の多様性の維持

GA における問題点の 1 つとして，過剰収束 (Premature Convergence) があげられる．これは，適応度の高い個体のもつ遺伝子が急速に集団中に広がることにより，集団中の遺伝子の均一化がおこり，進化の停滞を招く問題である．

集団遺伝学における知見として，2 つの集団が地理的に隔離されて遺伝子の交換が妨げられることによって生殖的な隔離に至る，地理的種形成 (Geographic Speciation) というものがある．

GA の並列化においても，複数の集団の同時探索，集団間の遺伝子の交換の制御によって各々の集団が異なる遺伝子構成に進化することが期待でき，全体として遺伝子の多様性の維持をはかることができる．

2.2.2 並列 GA の分類

GA は並列度の高いアルゴリズムであることから，研究の初期から並列規模，アーキテクチャに応じた多くの手法が提案されてきている．[Gordon 93, Cantú-Paz 97] では，それらの手法を 3 種に分類している．

大域並列 GA

大域並列 GA (Global parallel GA) は，1 つの Master PE と複数の Slave PE で構成される．Master PE が解集団を管理し，選択，交叉をおこなう．Slave PE は Master PE から送られて来た個体の突然変移，適応度計算をおこない，Master PE に返す．適応度計算の計算コストが高い問題ほど並列効果は高くなる．

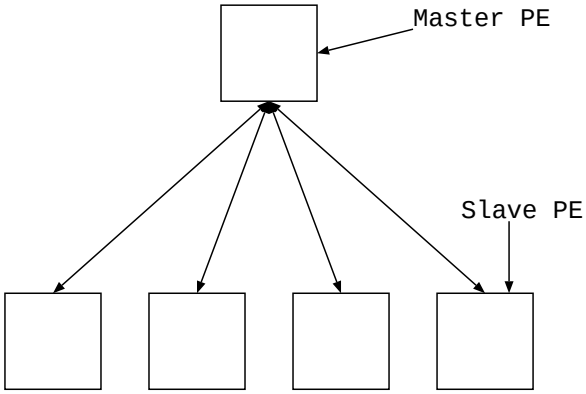

図 2.1: 大域並列 GA

島モデル並列 GA

島モデル並列 GA (Island Model Parallel GA) は粗粒度並列 GA (Coarse-Grained Parallel GA)，分散 GA (Distributed GA) とも呼ばれる．集団を複数の部分集団 (Subpopulation) に分割し，並列計算機上の各 PE に部分集団を割り当てる．一定期間ごとに移住 (Migration) と呼ばれる操作により，部分集団間で遺伝子の交換をおこない，集団の過剰収束を回避し，多様性の維持をはかる．島モデル並列 GA は，各 PE を隔離された孤島とみなせば，集団遺伝学における地理的種形成や，遺伝的浮動のアナロジーによって理解することができる．

[Grefenstette 81] において “interesting questions” として,「移住の頻度」,「移住相手の選択法」,「移住が過剰収束回避にあたえる影響」が指摘されているように，島モデル並列 GA では，移住の方法が最も重要な検討課題である．

「移住の頻度」については，一定期間ごとに各部分集団が同時に移住をおこなう同期型，任意の条件によって各部分集団が各々独立して移住をおこなう非同期型に分けられる．さらに，非同期型の場合，何をトリガーに移住操作を起動するかが問題になってくる.「移住相手の選択方法」については，隣接する PE から移住者を受け入れる近傍モデル，ランダムに移住元 PE を選択するモデル，自分とは異なる遺伝子構成の部分集団を選択するモデルがある．

大規模並列 GA

大規模並列 GA (Massively Parallel GA) は，細粒度並列 GA (Fine-Grained Parallel GA) とも呼ばれる．各 PE が 1 つの遺伝子をもち，近傍の PE と交叉をおこなう．このモデルの特徴として，交叉が近傍の PE に制限されることによって選択圧力が緩和され，多様性の維持がはかれる点があげられる．逆に単純な問題では探索効率が低下することもある．

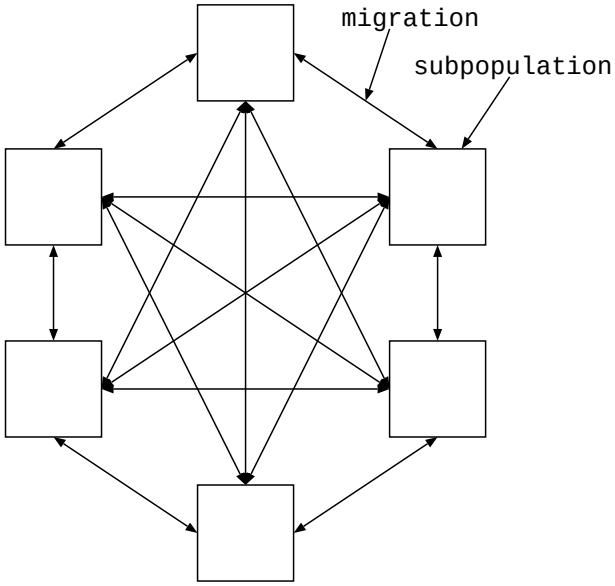

図 2.2: 島モデル並列 GA

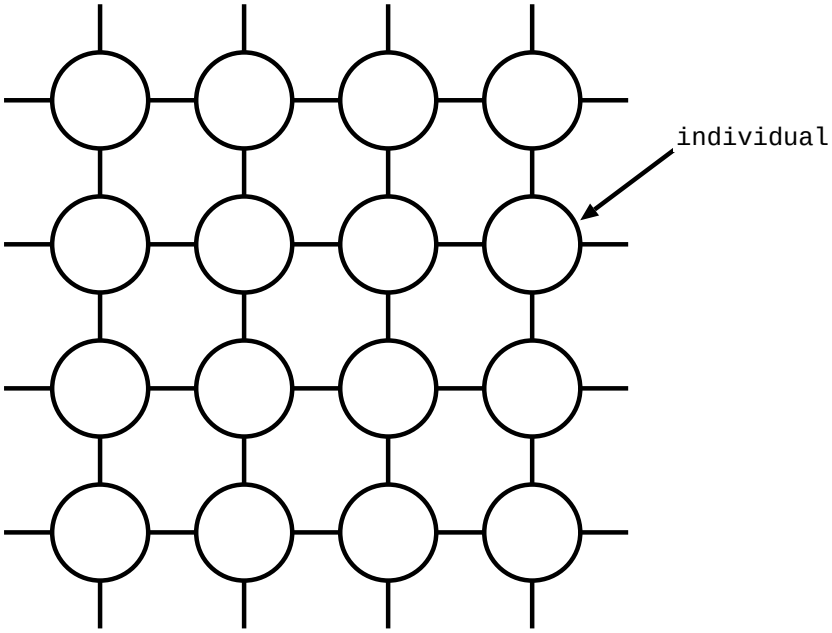

図 2.3: 大規模並列 GA

第 3 章

非同期並列遺伝的アルゴリズム

本章は本研究の特徴，狙いについて述べる．本研究では，移住操作の枠組と異なる探索傾向をもった部分集団群による協調探索を提案している．

3.1 移住操作

本研究で提案する並列化手法は島モデルに分類される．島モデル並列 GA において,「いつ」「だれと」移住操作をおこなうかが探索能力に大きな影響を与える．[Tanese 89] では一定世代ごとに隣接する部分集団から一定割合の遺伝子の移住をおこなう分散 GA(Distributed GA) が提案されており，Walsh 多項式最適化問題に適用している．また [Belding 95] では分散 GA のロイヤル・ロード関数最適化問題への適用例も報告されている．一定期間ごとに移住をおこなう同期型アルゴリズムでの実験の結果 [堀井 96]，ある程度の期間独立して探索をおこない，各部分集団において探索が進み，集団中の遺伝子の多様性が失われた後に移住をおこなうことが効率的であることを確認している．よって本研究では非同期に移住をおこなうモデルを採用する．対象問題によって探索の進度が異なるだけでなく，各部分集団によっても進度が異なるため，各部分集団が各々の探索状況に応じて非同期に移住をおこなうことが有効であるといえる．また，部分集団内の遺伝子の多様性の維持のためには，無作為に移住相手を選ぶのではなく，自分とできるだけ異なる遺伝子構成を持つ部分集団から移住者を導入しなければならない．[Munetomo 93, 棟朝 94] では非同期に移住をおこなうモデルが提案されており，適応度分布の標準偏差の相対的な減少割合によって集団の一様化を検出し，移住操作の起動条件とし，移住相手の選択の指標として各部分

集団の適合度の分布の平均値と標準偏差を用いている．

3.1.1 探索状況の把握

各部分集団が各々の探索状況に応じて，自分と異なる遺伝子構成を持つ部分集団から移住者を受け入れることにより，効率的な移住をおこなうことが期待できるが，そこで何を尺度に探索状況を把握するかが問題になってくる．例えば [Munetomo 93, 棟朝 94] では，集団の一様化と個体の適合度分布の標準偏差の減少が同時に起こるとする仮説，Sigma-Hypothesis に基づき移住時期を決定し，移住相手の選択には各部分集団内の個体の適合度分布の平均値と標準偏差を利用している．しかしながら，組合せ最適化問題や多峰性関数のような，遺伝子の類似度と適合度の類似度が一致しないような問題においては，適合度分布の平均値，標準偏差によって探索状況を正確に把握することは困難である．

3.1.2 バイアス，一時スキーマ

本研究では探索状況，集団の多様性の検出法としてバイアスを，移住相手の選択に一時スキーマ (Temporal Schema) を利用することを提案する．

t 世代における遺伝子集団 $P(t)$ を，各個体のストリングを行ベクトルとする次のようなマトリクスで表現する．

$$P(t) = (p_{ij}^t)$$

この時, t 世代におけるバイアス B^t, 一時スキーマ TS^t は以下のように定義される [筒井 94]．

バイアス B^t

個体数 N，遺伝子長 L の集団の多様性はバイアス B^t $(0.5 \leq B^t \leq 1.0)$ として計測することができる．

$$B^t = \frac{1}{N \times L} \sum_{j=1}^{L} \left[\left| \sum_{i=1}^{N} p_{ij}^t - \frac{N}{2} \right| + \frac{N}{2} \right]$$

例えば，次のような遺伝子構成をもつ集団 $P(t)$ では，バイアス $B^t = 0.775$ と計算される．

$$
\begin{aligned}
P(t) &= \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \\
\left|\sum_{i=1}^{N} p_{ij}^{t} - \frac{N}{2}\right| - \frac{N}{2} &= \begin{pmatrix} 3 & 5 & 4 & 4 & 3 & 4 & 4 & 4 \end{pmatrix} \\
B^t &= 0.775
\end{aligned}
$$

一時スキーマ TS^t

一時スキーマTS^tは，長さLのストリングであり，t世代において集団中の遺伝子の各遺伝子座が，‘1’，‘0’のどちらに収束しているかを示している．

$$
TS^t = (ts_1^t, ts_2^t, ..., ts_L^t)
$$

$$
ts_i^t = \begin{cases} 1 : \sum_{j=1}^{N} p_{ij}^t \geq N \times K_{TS} \\ 0 : \sum_{j=1}^{N} (1 - p_{ij}^t) \geq N \times K_{TS} \\ * : otherwise \end{cases}
$$

ここで，K_{TS}は任意に設定できる一時スキーマ検出しきい値であり，各遺伝子座においてK_{TS} $(0.5 < K_{TS} \leq 1)$以上の割合の個体が‘1’，または‘0’を持つ時，その遺伝子座は収束したとみなす．

一時スキーマの計算例を以下に示す．

$$
\begin{aligned}
P(t) &= \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \\
TS^t &= \begin{pmatrix} * & 1 & 0 & 0 & * & 1 & 0 & 1 \end{pmatrix}
\end{aligned}
$$

$K_{TS} = 0.8$ の場合

3.1.3 本研究における移住操作の枠組

本研究ではバイアス B^t が，しきい値 K_B を越えた時，部分集団中の遺伝子の多様性が失われたと判断し，移住をおこなう．移住相手の選択法として，一時スキーマ TS^t によって各部分集団の遺伝子構成を把握し，自分の一時スキーマと，他の部分集団の一時スキーマとのハミング距離を計測することによって，自分と異なる遺伝子構成を持つ部分集団を判断し，その部分集団から移住者を受け入れる．

自分と異なる遺伝子構成を持つ部分集団から移住者を受け入れれば，移住相手が収束しているならば，部分集団中の遺伝子の多様性を回復させることができるだけでなく，自分の持っていない良好なスキーマを導入することが期待できる．

3.2 多様な部分集団群による協調探索

島モデル並列 GA では集団を部分集団に分割し, 各部分集団が独立して探索をおこなう．小さな部分集団に分割されるため, 遺伝的浮動の進化に与える影響が大きくなり, 同じパラメータ設定であっても, 各部分集団の遺伝子構成は様々なものとなる．よって島モデルは暗黙の内に多様な部分集団群による協調探索となっている．

本研究では各部分集団の遺伝的操作のパラメータ設定を異なるものとし, 明示的に多様な部分集団となるように設定することを提案する．[Ishikawa 94] では 100%, 70%, 40%, 10% の世代ギャップが設定された 4 つの部分集団での協調探索が試みられている．ここでは適応度の高い個体の周辺を重点的に探索する部分集団と, 新しい個体を供給する部分集団との協調探索がおこなわれている．このような各部分集団に異なる探索傾向をもたせることによる役割分担は対象問題に対するロバスト性が期待できる．

第 4 章

実験 1 : ナップザック問題への適用

本章ではナップザック問題を用いた実験結果を示す.

4.1 ナップザック問題

ナップザック問題は制限重量を制約条件とし，重量と価値をパラメータとしてもつ荷物をナップザックに詰め込み，価値の総和が最大になる荷物の組合せを求める，組合せ最適化問題である．N 個の荷物があり，荷物 jの重量を w_j，価値を v_j とし，ナップザックの制限重量を W とする．荷物 j をナップザックに詰めるとき $x_j = 1$，詰めないとき $x_j = 0$ とすると，次のように定式化される．

$$\left.\begin{array}{ll} maximize & \sum_{j=1}^{N} v_j x_j \\ subject \quad to & \sum_{j=1}^{N} w_j x_j \leq W \\ & x_j \in \{0,1\}, j = 1, ..., N \end{array}\right\}$$

4.2 実験方法

実験 1 ではバイアス B^t，一時スキーマ TS^t によって探索状況を把握し，非同期に移住をおこなう非同期型並列 GA (Asynchronously Parallel GA:APGA) と，一定世代ごとに移住をおこなう同期型並列 GA (Synchronously Parallel GA:SPGA) との比較をおこなった.

CRAY T3E 上に実装し，個体総数を 512 として，部分集団数 16 ×個体数 32，および部分集団数 32×個体数 16 に分割した.

同期型並列 GA の移住間隔 (migration interval) は 20 世代とし，隣接する部分集団から移住者を受け入れることとした．

非同期型並列 GA はしきい値 $K_B = 0.95$ とし，バイアス $B^t \geq K_B$が 20 世代以上続いた時に移住操作が起動されるものとし，自分の一時スキーマ TS^t とのハミング距離がもっとも大きい一時スキーマを持つ部分集団から移住者を受け入れることとした．ただし，一時スキーマの内，$ts_i^t = *$ となるものは，$ts_i^t = 0.5$ として計算をおこなった．

テスト問題として 6 つのナップザック問題を扱い，問題 1,2,3 を荷物数 200，問題 4,5 を荷物数 300，問題 6 を荷物数 500 とし，各問題に対し，10 回の試行を 1000 世代までおこなった．

4.3 実験結果および考察

図 4.1〜4.6に問題 1〜6 の探索経過を示す．また，図 4.7，4.8に非同期型並列 GA，同期型並列 GA のバイアスの変化の代表例を示す．

探索初期は同期型並列 GA が非同期型並列 GA よりも早く良い解を得ているが，世代を経るにしたがって逆転し，非同期型並列 GA が同期型並列 GA よりも良い解を得ている．これは図 4.7，4.8のバイアスの変化が示すように，非同期型並列 GA は自分とできるだけ異なる遺伝子構成をもつ部分集団から移住者を受け入れているため，多少探索効率が低下するが，集団中の遺伝子の多様性を維持することができ，局所解への収束が回避しやすく，一方同期型並列 GA は，無条件に隣接する部分集団から移住者を受け入れるために，探索初期は移住によって多様性を維持されているが，次第に全部分集団の遺伝子構成が一様化していき，移住の効果が失われているからである．

部分集団数 32 の時の方が部分集団数 16 の時よりも非同期型 GA と同期型 GA の探索能力の差が大きいが，これは個体総数を固定していることにより，部分集団数が多いほど各部分集団のもつ個体数が少なるために，遺伝的浮動 (random genetic drift) の影響が強くなり，異なる遺伝子構成をもつ部分集団からの移住者の導入による多様性の維持の効果が大きくなるからである．

また，非同期型 GA の方が同期型 GA よりも部分集団数の増加による探索能力の低下が比較的緩やかなのは，各部分集団の個体数の減少による探索能力の低下を遺伝的浮動の効果が補っているからである．

以上の結果より，本研究で提案する移住方法による非同期型並列 GA は同期型並列 GA

と比較してスケーラビリティの向上がはかられたと言える.

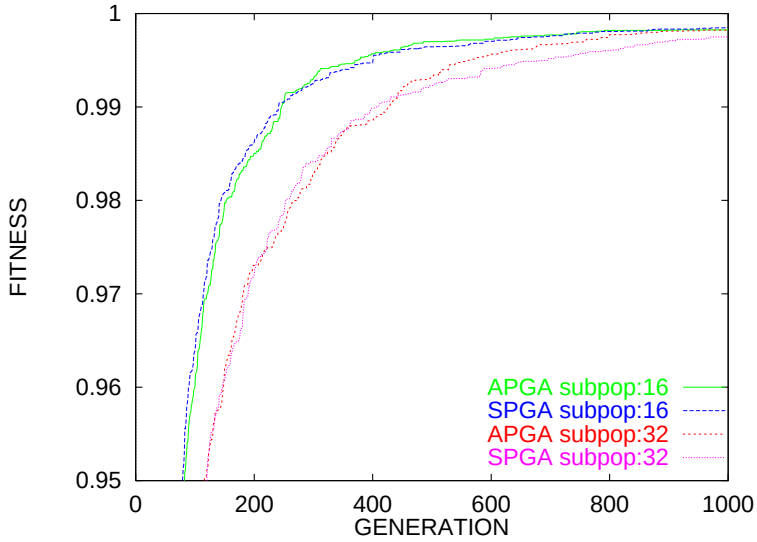

図 4.1: 問題 1 (荷物数:200)

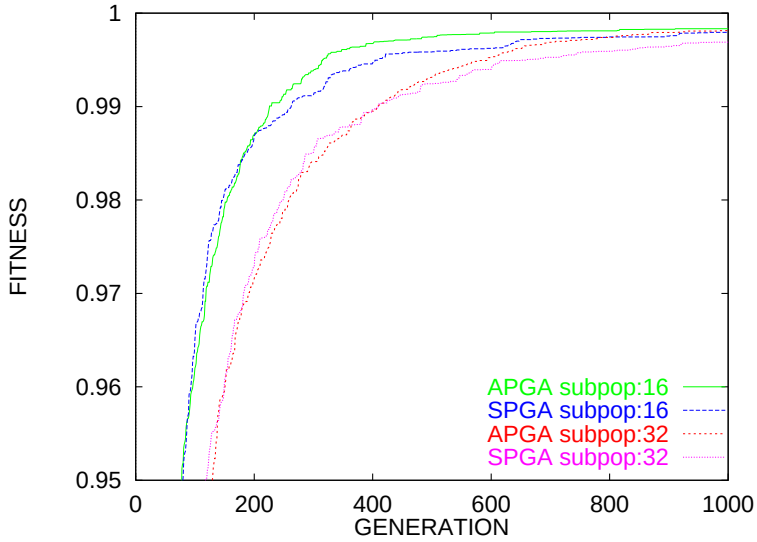

図 4.2: 問題 2 (荷物数:200)

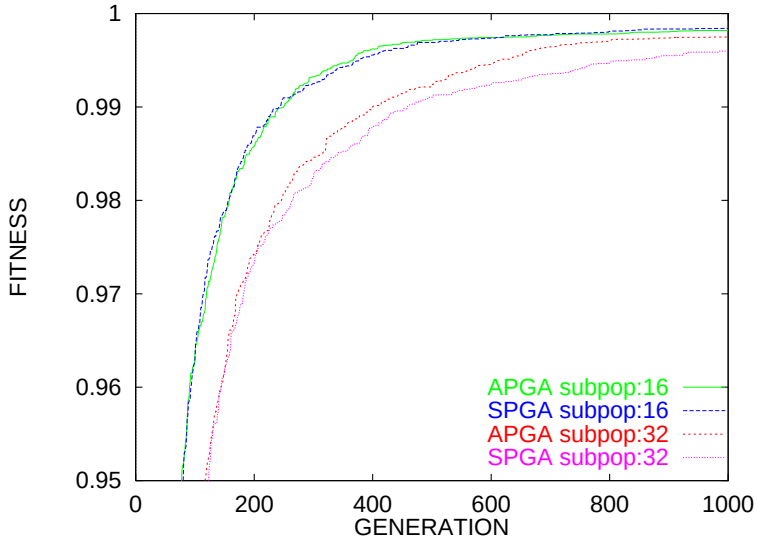

図 4.3: 問題 3 (荷物数:200)

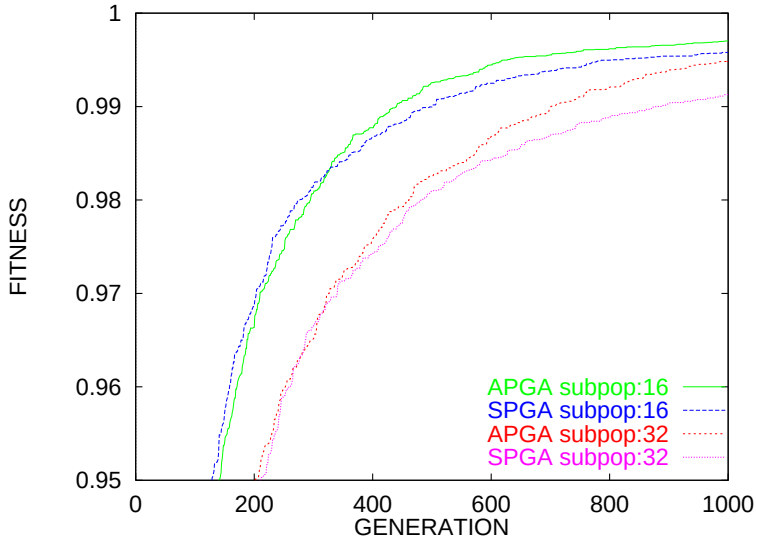

図 4.4: 問題 4 (荷物数:300)

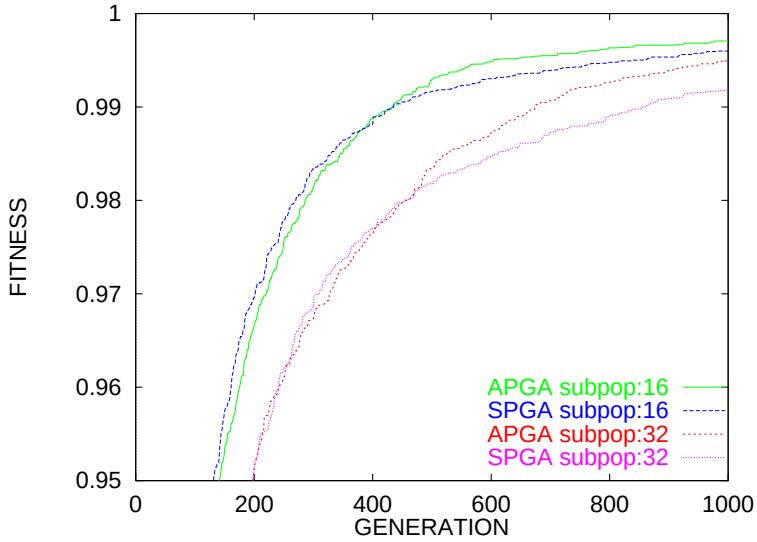

図 4.5: 問題 5 (荷物数:300)

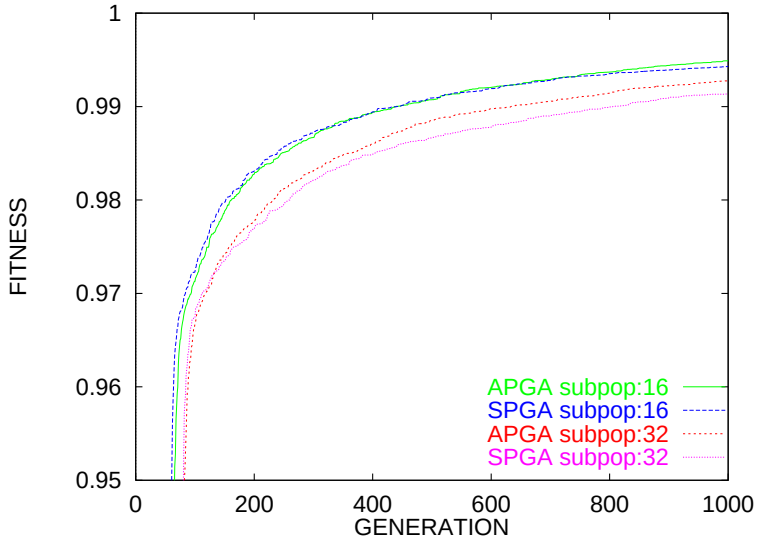

図 4.6: 問題 6 (荷物数:500)

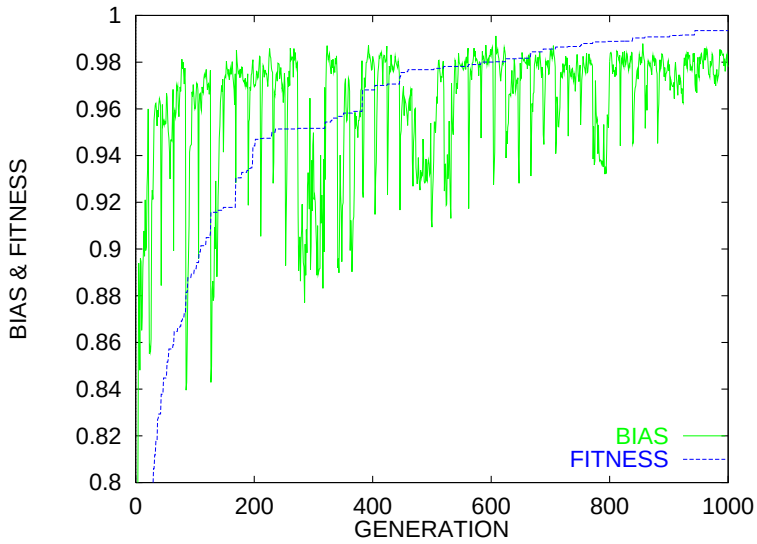

図 4.7: APGA のバイアスの変化

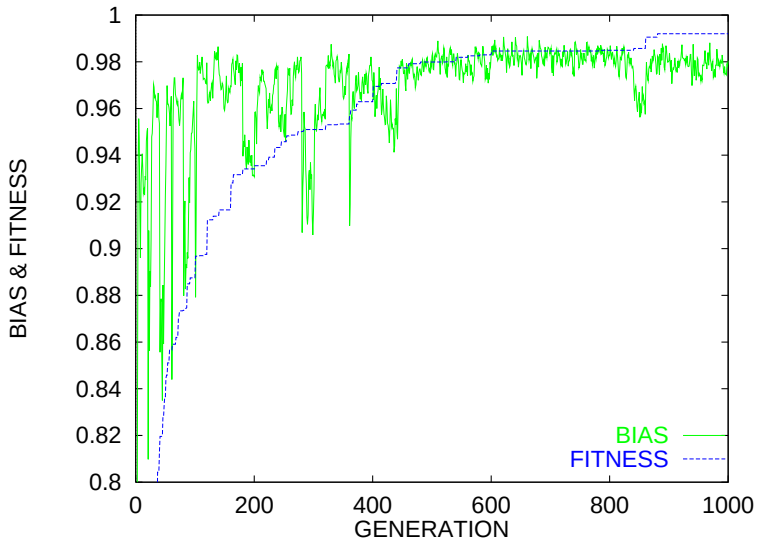

図 4.8: SPGA のバイアスの変化

第 5 章

実験 2 : ロイヤル・ロード関数への適用

本章ではロイヤル・ロード関数を用いた実験結果を示す.

5.1 ロイヤル・ロード関数

GA は積木仮説 (Building Block Hypothesis) を前提にした最適化アルゴリズムである. 積木仮説とは積木が積み上げられることによって目的とする物が組み上げられるように, 適応度向上に寄与する短いスキーマが交叉によって合成され, 高次のスキーマが形成されるプロセスの繰り返しによって, 最適解が生成されるという仮説である.

ロイヤル・ロード関数 (Royal Road Function) は, Mitchell らによって提案された評価関数である. この関数は積木仮説の GA における役割を詳細に解析するために, 積木仮説の特徴を明示的に与えられている. R_1, R_2[Mitchell 92, Forrest 93], R_3, R_4[Mitchell 94] の 4 種の関数が提案されている.

ロイヤル・ロード関数 R_1

R_1 は, 64 ビットのビット列で構成される. 8 つの 8 次のスキーマ s_1 〜 s_8 が積木として与えられ, 各スキーマ s_i は 係数 $c_i = order(s_i)$ をもつ. ビット列 x がそれぞれのスキーマのインスタンスであるとき, 係数 c_i が適応度に加算される. R_1 は, 次式のように定式化することができる.

図 5.1: ロイヤル・ロード関数 R_1

$$
\begin{aligned}
&R_1(x) = \textstyle\sum_{i=1}^{8} c_i \delta_i(x) \\
&\text{ただし} \quad c_i = order(s_i), \\
&\qquad\qquad \delta_i(x) = \begin{cases} 1 & (x \in s_i) \\ 0 & \text{otherwise} \end{cases} \\
&\text{例えば} \quad R_1(1111111100...0) = 8 \\
&\qquad\qquad R_1(1111111100...011111111) = 16 \\
&\qquad\qquad R_1(11111111...1) = 64
\end{aligned}
$$

ロイヤル・ロード関数 R_2

R_2 は R_1 を拡張したもので，64 ビットのビット列で構成される．R_1 と同様の 8 つの 8 次のスキーマ s_1 〜 s_8 と，それらの組み合わせで形成される 4 つの 16 次のスキーマ s_9 〜 s_{12} および 2 つの 32 次のスキーマ s_{13}, s_{14} が積木として与えられる．R_2 は，次式のように定式化することができる．

$$
\begin{aligned}
&R_2(x) = \textstyle\sum_{i=1}^{14} c_i \delta_i(x) \\
&\text{ただし} \quad c_i = order(s_i), \\
&\qquad\qquad \delta_i(x) = \begin{cases} 1 & (x \in s_i) \\ 0 & \text{otherwise} \end{cases} \\
&\text{例えば} \quad R_2(1111111100...011111111) = 16 \\
&\qquad\qquad R_2(11111111111111100...0) = 32 \\
&\qquad\qquad R_2(11111111...1) = 192
\end{aligned}
$$

```
s1 =11111111********************************************************;c1 =8
s2 =********11111111****************************************************;c2 =8
s3 =****************11111111************************************************;c3 =8
s4 =************************11111111********************************;c4 =8
s5 =********************************11111111**********************;c5 =8
s6 =****************************************11111111****************;c6 =8
s7 =************************************************11111111********;c7 =8
s8 =********************************************************11111111;c8 =8
s9 =1111111111111111************************************************;c9 =16
s10=****************1111111111111111********************************;c10=16
s11=********************************1111111111111111****************;c11=16
s12=************************************************1111111111111111;c12=16
s13=11111111111111111111111111111111********************************;c13=32
s14=********************************11111111111111111111111111111111;c14=32
```

図 5.2: ロイヤル・ロード関数 R_2

```
Level 1: s1 s2  s3 s4  s5 s6  s7 s8
Level 2:(s1 s2)(s3 s4)(s5 s6)(s7 s8)
Level 3:(s1 s2  s3 s4)(s5 s6  s7 s8)
Level 4:(s1 s2  s3 s4  s5 s6  s7 s8)
```

図 5.3: ロイヤル・ロード関数 R_3

ロイヤル・ロード関数 R_3

R_3 は R_1 と同様の 8 つの 8 次のスキーマ s_1 〜 s_8 および，R_2 と同様のそれらの組み合わせで形成されるスキーマを求める関数である．また， s_1 〜 s_8 の各スキーマの間にはイントロン (Intron) が挿入されている．8 次のスキーマをもつ個体をレベル 1 とし，より高次なスキーマをもつごとに，レベルが上昇する．レベル i の個体の適応度は $1.0 + i \times u$ となる．[Mitchell 94] では，$u = 0.2$ とされており，例えば，レベル 1 の個体の適応度は $1.0 + 1 \times 0.2 = 1.2$ となり，レベル 2 の個体の適応度は $1.0 + 2 \times 0.2 = 1.4$ となる．

ロイヤル・ロード関数 R_4

R_4 は 16 個の 8 次のスキーマ s_1 〜 s_{16} および，R_2 と同様のそれらの組み合わせで形成されるスキーマを求める関数である．R_3 と同様の適合度計算をおこなうが，各スキーマの間にはイントロンは存在しない．よって 128 ビットのビット列で 1 個体が表現される．

```
Level 1: s1 s2  s3 s4  s5 s6  s7 s8  s9 s10  s11 s12  s13 s14  s15 s16
Level 2:(s1 s2)(s3 s4)(s5 s6)(s7 s8)(s9 s10)(s11 s12)(s13 s14)(s15 s16)
Level 3:(s1 s2  s3 s4)(s5 s6  s7 s8)(s9 s10  s11 s12)(s13 s14  s15 s16)
Level 4:(s1 s2  s3 s4  s5 s6  s7 s8)(s9 s10  s11 s12  s13 s14  s15 s16)
```

図 5.4: ロイヤル・ロード関数 R_4

5.2 実験 2.1 : 移住方法の検証

本節では積木仮説の特徴をもった問題における非同期型並列 GA (以下 APGA) の効果を検証した.

5.2.1 実験方法

CRAY T3E 上に実装し，個体総数を 1024 として，部分集団数 4 × 個体数 256，部分集団数 8 × 個体数 128，部分集団数 16 × 個体数 64，部分集団数 32 × 個体数 32 の 4 通りのアルゴリズム，および個体数 1024 の単純 GA(SGA) の比較をおこなった．各問題に対し，30 回の試行を R_1, R_2 は 200 世代，R_3, R_4 は 2000 世代までおこなった.

5.2.2 実験結果および考察

ロイヤル・ロード関数 R_1 〜 R_4 における探索経過を 5.5 〜 5.8 に示す.

全ての結果において, 並列度が低い方が収束が早い. 並列度が高くなるほど部分集団の個体数が小さくなるため, 遺伝的浮動の影響が大きくなり, 部分集団の遺伝的構成はより多様なものとなる. よって集団全体としての収束は遅くなり, 探索効率は低下する. そのため問題が単純な場合には, このような手法は適さない.

しかし, R_3,R_4 のような困難な問題では, 並列度が高くなるほど集団の多様性の維持によって局所解に陥りにくくなり, 最終的にはより良い解が得られている.

また, ナップザック問題では並列度を高めると解精度が低下しているが, ロイヤル・ロード関数では解精度が向上している. これは, ロイヤル・ロード関数が積木仮説が理想的に成り立つ問題であり, 島モデルにおいては遺伝的浮動によって部分集団ごとに異なる積木を同時並行に探索できるからである. 各部分集団が異なる積木を発見し, 他の集団との移住による相互補完によって, より良い解が発見しやすくなっているのだと考えられる. よっ

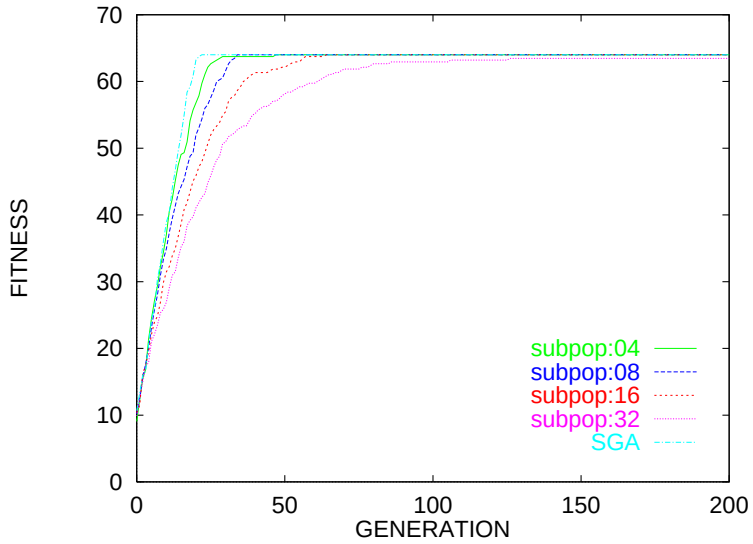

図 5.5: ロイヤル・ロード関数 R_1

て積木仮説が成立する傾向の強い問題, たとえば巡回セールスマン問題や, 実用的な問題としては, LSI の回路設計のような問題に適した手法であるといえる.

5.3 実験 2.2 : 異質な部分集団群による探索の検証

本節では, 非同期型並列 GA (APGA) において明示的に多様な部分集団群となるように遺伝的操作のパラメータとして, 異なるパラメータを各部分集団に設定した場合の検証をおこなう.

5.3.1 実験方法

CRAY T3E 上に実装し, 個体総数を 1024 として, 部分集団数 32 × 個体数 32 に分割した. APGA における各部分集団に同じパラメータを与えたモデル (Single Parameter) と, 異なるパラメータを与えたモデル (Multiform Parameters) との比較をおこなった.

Single Parameter のパラメータ設定は, 世代ギャップ 0.8 , 突然変移率 0.01, 移住操作が起動するしきい値 K_B を 0.9 とした.

また, Multiform Parameters のパラメータ設定は, (世代ギャップ, 突然変移率) を (0.8,0.01), (0.5,0.05) の 2 通りに, 移住操作が起動するしきい値 K_B を 0.9,0.8 の 2 通りに設定した.

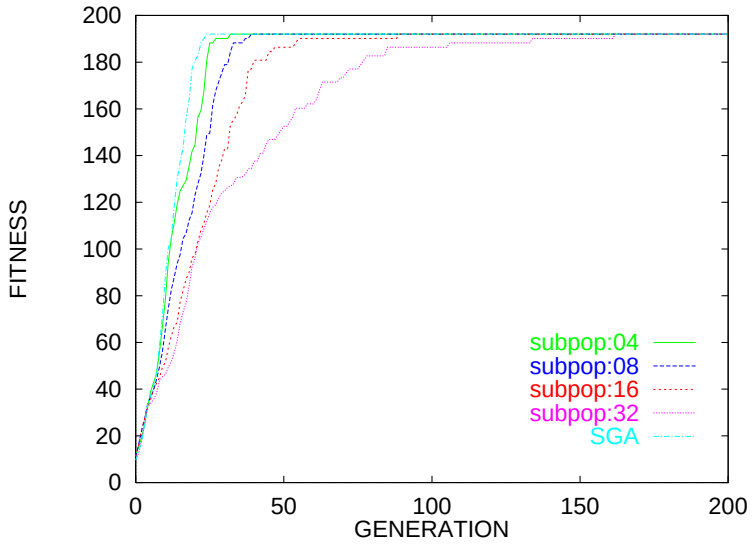

図 5.6: ロイヤル・ロード関数 R_2

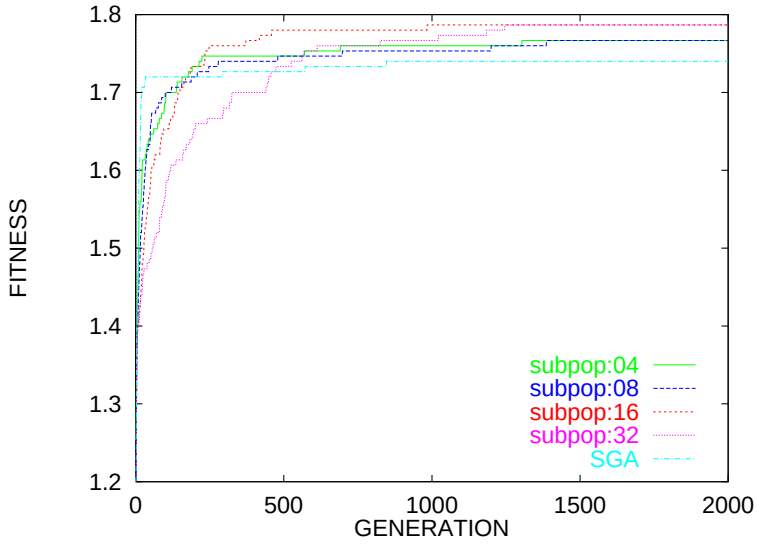

図 5.7: ロイヤル・ロード関数 R_3

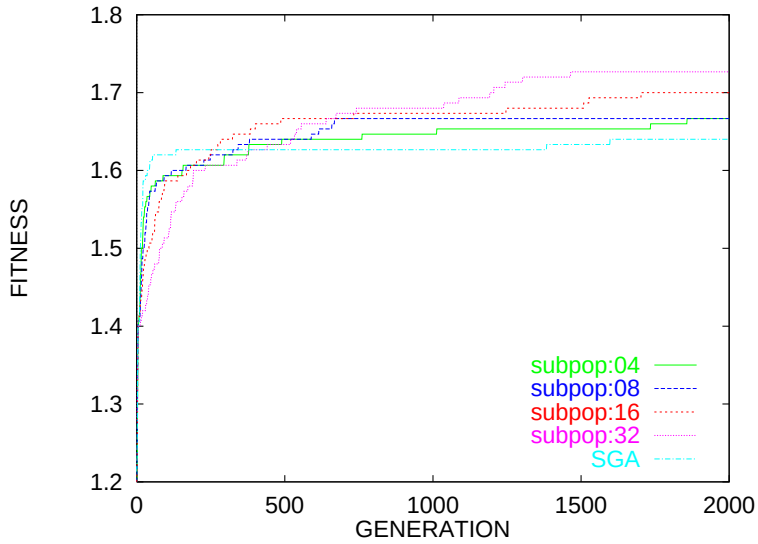

図 5.8: ロイヤル・ロード関数 R_4

よって各部分集団のパラメータ設定は {(世代ギャップ, 突然変移率),K_B} が {(0.8,0.01),0.9}, {(0.8,0.01),0.8}, {(0.5,0.05),0.9}, {(0.5,0.05),0.8} の 4 通りの組合せとなり，各パラメータ設定の組が 8 つの部分集団をもつ.

以上のパラメータ設定によって，ロイヤル・ロード関数 R_3, R_4 に対し，30 回の試行を 2000 世代までおこなった.

5.3.2 実験結果および考察

図 5.9, 5.10にロイヤル・ロード関数 R_3, R_4の探索経過を示す.

パラメータ設定において, (世代ギャップ, 突然変移率) が (0.8,0.01) の集団では, 交叉に重点を置いた探索, (0.5,0.05) の集団では, 突然変移に重点を置いた探索がおこなわれる. また, $K_B = 0.9$ の集団では, 自分の集団内での十分な探索の後に移住をおこなっているが, $K_B = 0.8$ の集団では, 他集団からの移住者の導入によって, 大域的な探索に重点を置いている. このため, Multiform Parameters は Single Parameter と比較して探索効率は低下しているが, 集団全体の遺伝子の多様性が保存されやすく, 局所解に陥りにくいと考えられる.

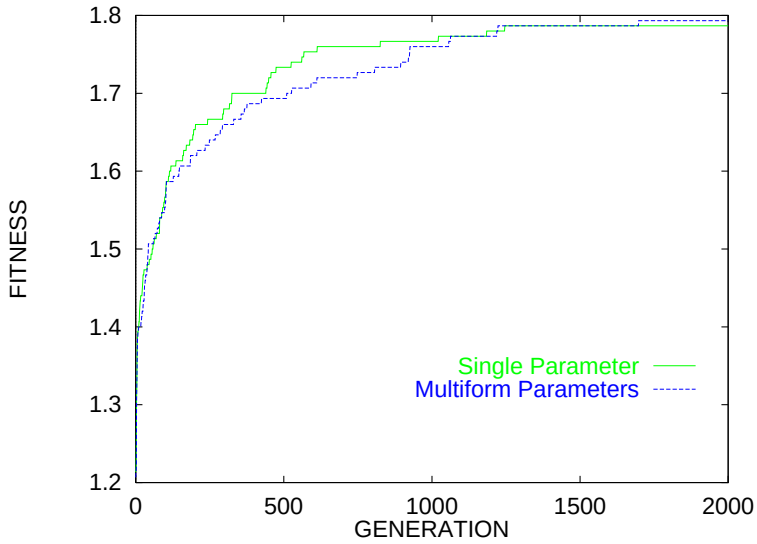

図 5.9: ロイヤル・ロード関数 R_3

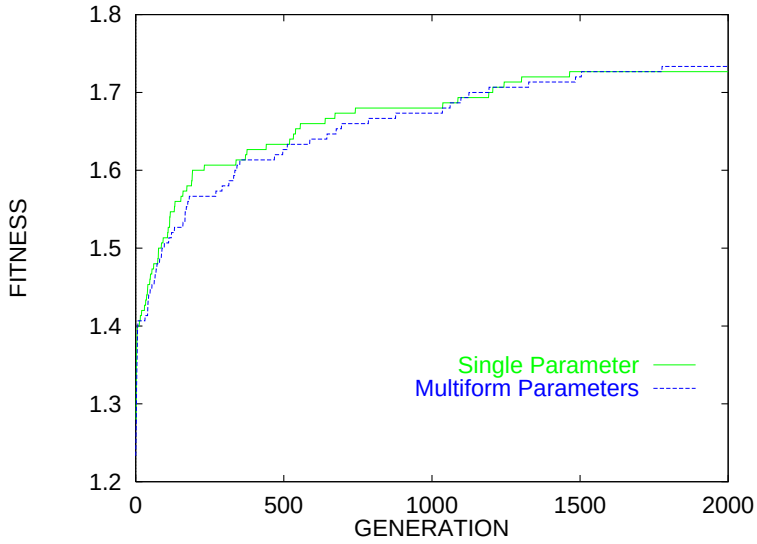

図 5.10: ロイヤル・ロード関数 R_4

第 6 章

おわりに

本研究では, 各部分集団が各々の探索状況に応じて非同期に移住操作をおこなう非同期並列 GA (Asynchronously Parallel GA) において, 以下の提案をおこなった.

- 部分集団内の遺伝子の一様化を移住操作の起動条件とし, 一様化の検出方法としてバイアスを利用する.
- 自分と異なる遺伝子構成をもつ部分集団からの移住者の導入を提案し, 部分集団の遺伝子構成を把握する指標として一時スキーマを利用する.
- 各部分集団に異なる遺伝的操作のパラメータを設定することによって, 明示的に異なる探索傾向をもたせる.

以上の提案をナップザック問題, ロイヤル・ロード関数に対して適用し, その効果を検証した結果, 以下の点が明らかとなった.

- 本研究で提案した移住操作の枠組は, 組合せ最適化問題において効果的である. 特に積木仮説が成立する傾向の強い問題においては集団の分割による積木の並列探索によってより効果的な探索が可能になる.
- 各部分集団が異なる探索傾向をもって探索した場合, 探索効率は低下するが, 集団全体の遺伝子の多様性の維持によって探索精度は向上する.

今後の研究課題としては, 本アルゴリズムの詳細な解析, および巡回セールスマン問題のようなより複雑な問題への適用による性能評価が必要であると考える.

謝辞

本研究を遂行するにあたり，多大なる御指導を賜わりました指導教官の國藤進教授に心から御礼申し上げます.

また，研究生活の様々な面において支援して頂きました國藤研究室の皆様に心から御礼申し上げます.

最後に，両親，祖父母の物心両面の援助によって研究に専念できたことに対し心から感謝致します.

1998 年 2 月 13 日

堀井宏祐

参考文献

[Baker 85] J. E. Baker, Adaptive Selection Methods for Genetic Algorithms, in *Proceedings of the First International Conference on Genetic Algorithms*, pp.101-111, 1985.

[Belding 95] T. C. Belding, The Distributed Genetic Algorithm Revisited, in *Proceedings of the Sixth International Conference on Genetic Algorithms*, pp.114-121, 1995.

[Cantú-Paz 97] E. Cantú-Paz, A Survey of Parallel Genetic Algorithms, IlliGAL Report No.97003, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, 1997.

[Forrest 93] S. Forrest, M. Mitchell, Relative Building-Block Fitness and the Building-Block Hypothesis, in *Foundations of Genetic Algorithms 2*, pp.109-126,1993.

[Goldberg 89] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, 1989.

[Gordon 93] V. S. Gordon, and D. Whitley, Serial and Parallel Genetic Algorithms as Function Optimizers, in *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp.177-183, 1993.

[Grefenstette 81] J. J. Grefenstette, Parallel adaptive algorithms for function optimization, Technical Report No.CS-81-19, Computer Science Department,Vanderbilt University, 1981.

[Ishikawa 94] M. Ishikawa,T. Toya, and Y. Totoki, Parallel Application Systems in Genetic Processing, in *Proceedings of International Symposium on Fifth Generation*

Computer Systems, 1994.

[Michalewicz 92] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, 1992.

[Mitchell 92] M. Mitchell, S. Forrest, and J. H. Holland, The Royal Road for Genetic Algorithms: Fitness Landscapes and GA Performance, in *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, pp.245-254,1992.

[Mitchell 94] M. Mitchell, J. H. Holland, and S. Forrest, When Will a Genetic Algorithm Outperform Hill Climbing?, in *Advances in Neural Information Processing 6*, 1994.

[Munetomo 93] M. Munetomo, Y. Takai, and Y. Sato, An Efficient Migration Scheme for Subpopulation-Based Asynchronously Parallel Genetic Algorithms, in *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp.649, 1993.

[Nimwegen 97] E. van Nimwegen, J. P. Crutchfield, and M. Mitchell, Statistical Dynamics of the Royal Road Genetic Algorithm, Santa Fe Institute Working Paper 97-04-035, 1997.

[Syswerda 89] G. Syswerda, Uniform Crossover in Genetic Algorithms, in *Proceedings of the Third International Conference on Genetic Algorithms*, pp.2-9, 1989.

[Tanese 89] R. Tanese, Distributed Genetic Algorithm, in *Proceedings of the Third International Conference on Genetic Algorithms*, pp.434-439, 1989.

[北川 91] 北川修，集団の進化，東京大学出版会，1991.

[筒井 94] 筒井茂義，藤本好司，個体群探索分岐型遺伝的アルゴリズム fGA (Forking GA) の提案，人工知能学会誌, Vol.9, No.5, pp.741-747,1994.

[堀井 96] 堀井宏祐，小泉孝之，辻内伸好，國藤進，複数の異なる遺伝パラメータを持つ GA によるパラメータ感度低減化法の提案, 日本機械学会第 9 回計算力学講演会講演論文集, No.96-25, pp.1-2, 1996.

[堀井 97] 堀井宏祐，國藤進，松澤照男，島モデル並列 GA における部分集団間の相互作用, 人工知能学会研究会資料 SIG-FAI-9702, pp.13-18,Sep.,1997

[棟朝 94] 棟朝雅晴，高井昌彰，佐藤義治， 集団分割型非同期並列遺伝的アルゴリズムにおける個体交換アルゴリズムの改良と評価， 情報処理学会論文誌, Vol.35, No.9, pp.1815-1827, 1994.