



確率的探索法の関数最適化問題への適用に関する研究

足立, 進

(Degree)

博士 (工学)

(Date of Degree)

2001-04-27

(Date of Publication)

2014-11-11

(Resource Type)

doctoral thesis

(Report Number)

乙2545

(URL)

<https://hdl.handle.net/20.500.14094/D2002545>

※ 当コンテンツは神戸大学の学術成果です。無断複製・不正使用等を禁じます。著作権法で認められている範囲内で、適切にご利用ください。



神戸大学博士論文

確率的探索法の関数最適化問題
への適用に関する研究

2001年2月

足立 進

要旨

最適化手法には厳密解法と近似解法があるが、このうち、ほとんどの問題に対して工学的に実用的なものは近似解法である。近似解法は探索の仕方によって、探索方向が一意的に決定している確定的探索と、探索方向を確率的に決定する確率的探索に分けられる。確率的探索法の代表的な計算方法としては、進化論をベースとして考案された進化的計算法と、統計力学をベースとした統計力学的計算法があり、それぞれ独立した計算法として発展してきたという歴史を持つ。従来より、これらの計算法の融合による拡張などは議論されてきたが、両者がどのような関係にあるかという点については明らかにされていない。

本論文では、確率的探索アルゴリズムの総合的な評価のため、進化的計算法と統計力学的計算法の両者の関係を明らかにするとともに、優れた探索アルゴリズムがどのようなものであるかを考案した。優れた確率的探索アルゴリズムとは、汎用的で設計が容易なアルゴリズム、収束性能が優れたアルゴリズム、および探索における解の精度、あるいは成功率が高いアルゴリズムに分類されると考え、これらの目的に対する計算法を、遺伝的アルゴリズムをベースとしてそれぞれ構築し、性能評価を行った。これにより、特に汎用的な計算法の開発は、将来的には、汎用的なコンピュータの開発の基礎となる可能性があると考えられる。

まず汎用的で、かつ設計が容易であるという互いに矛盾した目的を同時に満足するような遺伝的アルゴリズムを考案した。解くべき問題の詳細な情報が得られない場合には、できるだけ汎用的なアルゴリズムを用いる方が望ましい。遺伝的アルゴリズムは本来、汎用的なアルゴリズムであるため、これを改良することにより目的を達成できると考えた。遺伝的アルゴリズムの要素は、個体集合、個体の適応度、個体に対する演算の3つからなり、非常に多くのパラメータが存在する。設計を容易にするためには、このパラメータの設定に費す労力を軽減することが必要である。そこで本論文では、パラメータの設定を必要としない遺伝的アルゴリズム(パラメータフリー遺伝的アルゴリズム)を提案した。このアルゴリズムの基本概念は、変則的な選択による可変な個体数と、非対称な突然変異である。これによって局所的探索と大域的探索のバランスをとりながら適応的に探索を行うアルゴリズムを構築することができた。アルゴリズムの能力を評価するために、9種類の関数最適化問題に適用し、従来法である定常状態遺伝的アルゴリズムと比較した。また、アルゴリズムの優劣を判断する基準として大域的探索性能を表す成功率と、局所的探索性能を表す平均評価回数を統計量として求めた。その結果、ほぼ全ての関数に対してパラメータフリー遺伝的アルゴリズムの収束性能と成功率の高さを確認した。すなわち、様々な関数に対して非常に汎用的であり、かつパラメータ設計に労力を費さなくて済むアルゴリズムであることが示された。

次に、収束性能を向上させるための拡張として、パラメータフリー遺伝的アルゴリズムの並列分散処理を行った。収束を速めるため、局所探索を重視する従来の設計法では、速やかに局所解に陥ってしまい、真の解に近付くことが困難となる。そこで、遺伝的アルゴリズム

の大域的探索能力をそのまま引き継ぎ、なおかつ収束性に優れたアルゴリズムを考案した。並列分散処理は、その効果が発揮されるのは特に探索問題に対してであり、また、複数の局所集団をそれぞれ並列に実行させ、局所集団間で個体の移民を行わせる島モデルを用いれば、容易に実装可能である。実装方法としては、局所集団の役割で2種類、すなわち一様分散型とマスタースレーブ型、また、移民の仕方でも2種類、すなわち直接移民型と階層移民型の合計4種類を考えた。実験の結果、一様分散/直接移民型の場合が最も収束性能が優れていることを確認した。また、部分的に成功率を増加させることもできた。これは、局所集団の増加によって多様性が増大した結果である。

解の精度を向上させる、あるいは等価的に成功率を向上させるための拡張として、遺伝子重複説に基づいた遺伝的アルゴリズムを考案した。成功率を向上させるためには個体の多様性が必要であるが、単に初期個体数を増加させるならば、これはパラメータ設定の問題となり、しかも個体数の増加は余分な評価回数を生じるため、与えられた時間内に探索できなくなる可能性がある。そこで優秀なスキームを組み合わせることで多様性を増大する方法を、遺伝子重複により実現した。遺伝子重複を用いるアルゴリズムは、短い遺伝子を低次元の解空間における個体に対応させ、これらを組み合わせることでより長い遺伝子を生成し、これを高次元の解空間における個体とみなして探索を行う方法である。この手法を、ベースとなるアルゴリズムにパラメータフリー遺伝的アルゴリズムと定常状態遺伝的アルゴリズムの2つを用い、遺伝子の重複方法として、連鎖型、伸長型、結成型、および拡張結成型の4つの実装方法で構成し、実験を行った。その結果、ほぼ全ての関数で重複の効果が現れ、ベースとなるアルゴリズムに無関係に最良の重複方法が関数ごとに存在することを確認した。また、概してベースとなるアルゴリズムはパラメータフリー遺伝的アルゴリズムの方が優れている結果となった。多峰性、非対称性、変数分離不可能性を同時に持つような関数に対しては、アルゴリズムの違いによる成功率の向上は現れなかったが、定常状態遺伝的アルゴリズムベースの場合は収束性能の向上が見られた。

次に、確率的探索法の総合的な評価のため、統計力学的計算法と進化的計算法との比較を行った。統計力学的計算法の代表としてモンテカルロ法、進化的計算法の代表として上記のパラメータフリー遺伝的アルゴリズムを用いた。両者は問題を有限の bit 列で表現できるという共通点があるため、同じ条件下で同じ問題を適用することによって具体的な比較が可能となると考えた。比較の結果、モンテカルロ法は収束性に優れているのに対し、成功率に関しては遺伝的アルゴリズムの方が優れていることが確認された。これは、モンテカルロ法が個体数1の遺伝的アルゴリズムに相当し、このために多様性が少ないのが原因である。また、それぞれのアルゴリズムの出発点の根底にある平衡、非平衡の概念についても触れた。

最後に遺伝的アルゴリズムの理論解析について述べた。従来より理論解析は、ごく単純化したモデルで、また、非常に簡単な目的関数に対してのみ成功しているだけであり、一般的な非線形関数に対しての解析は依然、困難である。このような困難さの背景には、解を探索する個体集団の存在があり、それは上記の非平衡の問題でもある。そこで、個体の集団的な振る舞いである分布関数を直接求めるのではなく、それぞれの遺伝的演算子の振る舞いを調べ、遺伝的アルゴリズムの改良のための解析を試みた。まず突然変異については、突然変異を受けることにより個体の移動距離の分散が突然変異率に比例するという定量的な結果を得た。また交叉については、親個体のハミング距離が大きいときは、一様交叉によって生成される子個体は全探索空間に広く分布し、逆に小さいときは、親個体の近傍に集中するこ

とが分かった。これによって、遺伝的アルゴリズムにおける交叉の重要性が確認された。続いて、本論文で用いた、アルゴリズムの評価基準である成功率と平均評価回数に対する考察を行った。これにより、これらの関係性と、並列処理における探索効率の改善などの議論がより明確となった。

目次

要旨	i
第1章 序論	1
1.1 研究の背景	1
1.2 研究の動機	1
1.3 研究の目的	3
1.3.1 汎用的なアルゴリズムの考案	3
1.3.2 探索を速くするための工夫	4
1.3.3 探索を成功させるための工夫	4
1.3.4 統計力学的計算法と進化的計算との比較	4
1.4 本論文の構成	4
1.5 従来法としての遺伝的アルゴリズム	6
1.5.1 背景と概要	6
1.5.2 コーディング	7
1.5.3 遺伝的オペレータ	8
1.5.4 スキーマ定理	9
1.5.5 従来手法 1: 単純遺伝的アルゴリズム	11
1.5.6 従来手法 2: 定常状態遺伝的アルゴリズム	12
第2章 パラメータフリー遺伝的アルゴリズムの提案と性能評価	17
2.1 概要	17
2.2 アルゴリズムの基本構造	17
2.2.1 局所集団, および家族	19
2.2.2 交叉, 突然変異	19
2.2.3 選択, 淘汰	20
2.3 実験 1: De Jong の F4	21
2.4 実験 2: ベンチマーク問題	22
2.5 実験 2 に対する結果	24
2.6 実験 3: 平均評価回数の評価と ICEO の他のアルゴリズムとの比較	30
2.7 実験 4: 成功率の導入と PfGA, SSGA の評価	30
2.8 考察: 選択における場合分け	31
2.9 まとめ	32

第 3 章	パラメータフリー遺伝的アルゴリズムの並列分散処理の提案と性能評価	35
3.1	概要	35
3.2	遺伝的アルゴリズムの並列化	35
3.2.1	アーキテクチャ: 一様分散型とマスタースレーブ型	36
3.2.2	移民方法: 直接移民型と階層移民型	37
3.3	4つの実装方法によるアルゴリズムの構築	37
3.4	実験と結果	39
3.4.1	Sphere model に対する結果	39
3.4.2	Double Sum に対する結果	42
3.4.3	Randomized Sphere model に対する結果	42
3.4.4	Randomized Double Sum に対する結果	42
3.4.5	Rastrigin's function に対する結果	42
3.4.6	Griewank's function に対する結果	51
3.4.7	Michalewicz' function に対する結果	51
3.4.8	Shekel's foxholes に対する結果	51
3.4.9	Langerman's function に対する結果	51
3.5	実験結果に対する考察と最良戦略	51
3.6	ICEO の他のアルゴリズムとの比較	60
3.7	まとめ	61
第 4 章	遺伝子重複説に基づいた遺伝的アルゴリズムの提案と性能評価	65
4.1	概要	65
4.2	遺伝子重複アルゴリズム	65
4.2.1	タイプ A (遺伝子連鎖型)	67
4.2.2	タイプ B (遺伝子伸長型)	68
4.2.3	タイプ C (遺伝子結合型)	68
4.2.4	タイプ D (遺伝子拡張結合型)	70
4.3	実験と結果	70
4.3.1	Sphere model に対する結果	70
4.3.2	Randomized Sphere model に対する結果	73
4.3.3	Michalewicz' function に対する結果	73
4.3.4	Randomized Double Sum に対する結果	73
4.3.5	Griewank's function に対する結果	79
4.3.6	Langerman's function に対する結果	79
4.4	実験結果に対する考察と最良戦略	79
4.5	ICEO の他のアルゴリズムとの比較	82
4.6	まとめ	83
4.6.1	ベースとなる GA に関して	84
4.6.2	遺伝子重複のタイプに関して	84
4.6.3	移民レートに関して	84
4.6.4	次元依存性に関して	84

4.6.5	並列分散処理との比較	84
第5章	モンテカルロ法による確率探索と遺伝的アルゴリズムとの比較	85
5.1	概要	85
5.2	モンテカルロ法のエネルギー関数最小化問題への適用に関する従来手法	85
5.2.1	イジングモデルのエネルギー最小化問題	85
5.2.2	イジングモデルの緩和過程のシミュレーション	87
5.2.3	シミュレーテッドアニーリングによる最適化	88
5.3	量子モンテカルロ法による最適化	89
5.3.1	量子スピン系を古典的スピン系に置き換える従来法	90
5.3.2	平均場近似を用いた量子モンテカルロ法	92
5.4	メトロポリス法の関数最適化問題への適用	96
5.4.1	アルゴリズム	96
5.4.2	実験と結果	98
5.5	考察: モンテカルロ法と遺伝的アルゴリズムとの比較	99
5.6	まとめ	101
第6章	考察	103
6.1	理論的考察	103
6.1.1	突然変異に対する理論解析	103
6.1.2	交叉に関する考察	106
6.2	評価基準に対する考察	107
第7章	結論	111
7.1	研究成果	111
7.1.1	パラメータフリー遺伝的アルゴリズム	111
7.1.2	並列分散処理への拡張	111
7.1.3	遺伝子重複による拡張	111
7.1.4	統計力学的計算法と進化的計算法との比較	112
7.1.5	理論解析	112
7.2	結論と今後の課題	113
	謝辞	115
	参考文献	117

表 目 次

1.1	binary coding と Gray coding との対応	8
2.1	第 1 回 ICEO と SSGA, PfGA の結果	31
2.2	SSGA と PfGA の SR と ENES	32
2.3	PfGA の選択における各ケースの割合	32
3.1	並列 PfGA(UD1) の最良戦略	61
3.2	第 1 回 ICEO と PfGA, 並列 PfGA の比較. (N) は並列度	62
4.1	PfGA と SSGA をベースとした遺伝子重複 GA の比較結果と最良戦略. (R) は最良戦略における移民レート.	82
4.2	第 1 回 ICEO と PfGA, GD-PfGA, および GD-SSGA の比較	83
5.1	モンテカルロ法と量子モンテカルロ法による成功率と ENES	99
5.2	ICEO5 関数に対するモンテカルロ法, 量子モンテカルロ法, および PfGA に よる成功率と ENES	99

目次

1.1	De Jong の F2 に対する個体の収束の様子 (SGA)	13
1.2	探索過程における個体の分布 (SGA)	13
1.3	De Jong の F2 に対する個体の収束の様子 (SSGA)	15
1.4	探索過程における個体の分布 (SSGA)	15
2.1	PfGA における集団の視点	18
2.2	inverse mutation, (a) $n_1 < n_2$ のとき. (b) $n_1 > n_2$ のとき.	20
2.3	PfGA における選択則	20
2.4	De Jong の F4 に対する PfGA の収束の様子	22
2.5	De Jong の F4 に対する PfGA の局所集団サイズの変化	22
2.6	Griewank's Function	25
2.7	Michalewicz' Function	25
2.8	Shekel's foxholes	25
2.9	Langerman's Function	25
2.10	5次元 Sp に対する, SSGA, および PfGA による最良個体の関数値	26
2.11	5次元 Ds に対する, SSGA, および PfGA による最良個体の関数値	26
2.12	5次元 Rs に対する, SSGA, および PfGA による最良個体の関数値	27
2.13	5次元 Rd に対する, SSGA, および PfGA による最良個体の関数値	27
2.14	5次元 Ra に対する, SSGA, および PfGA による最良個体の関数値	28
2.15	5次元 Gr に対する, SSGA, および PfGA による最良個体の関数値	28
2.16	5次元 Mi に対する, SSGA, および PfGA による最良個体の関数値	29
2.17	5次元 Sh に対する, SSGA, および PfGA による最良個体の関数値	29
2.18	5次元 La に対する, SSGA, および PfGA による最良個体の関数値	29
3.1	逐次探索 (a) と並列探索 (b)	36
3.2	一様分散型アーキテクチャ(a) とマスタースレーブ型アーキテクチャ(b). 矢印は個体の移動を表す	37
3.3	移民選択法: UD1(a), UD2(b), MS1(c), および MS2(d)	38
3.4	5次元 Sp に対する成功率	40
3.5	5次元 Sp に対する ENES	40
3.6	10次元 Sp に対する成功率	41
3.7	10次元 Sp に対する ENES	41
3.8	5次元 Ds に対する成功率	43
3.9	5次元 Ds に対する ENES	43

3.10	10次元Dsに対する成功率	44
3.11	10次元Dsに対する ENES	44
3.12	5次元Rsに対する成功率	45
3.13	5次元Rsに対する ENES	45
3.14	10次元Rsに対する成功率	46
3.15	10次元Rsに対する ENES	46
3.16	5次元Rdに対する成功率	47
3.17	5次元Rdに対する ENES	47
3.18	10次元Rdに対する成功率	48
3.19	10次元Rdに対する ENES	48
3.20	5次元Raに対する成功率	49
3.21	5次元Raに対する ENES	49
3.22	10次元Raに対する成功率	50
3.23	10次元Raに対する ENES	50
3.24	5次元Grに対する成功率	52
3.25	5次元Grに対する ENES	52
3.26	10次元Grに対する成功率	53
3.27	10次元Grに対する ENES	53
3.28	5次元Miに対する成功率	54
3.29	5次元Miに対する ENES	54
3.30	10次元Miに対する成功率	55
3.31	10次元Miに対する ENES	55
3.32	5次元Shに対する成功率	56
3.33	5次元Shに対する ENES	56
3.34	10次元Shに対する成功率	57
3.35	10次元Shに対する ENES	57
3.36	5次元Laに対する成功率	58
3.37	5次元Laに対する ENES	58
3.38	10次元Laに対する成功率	59
3.39	10次元Laに対する ENES	59
4.1	タイプAにおける遺伝子重複のイメージ	66
4.2	$N = 5$ の時のタイプA(遺伝子連鎖型)における遺伝子重複法	67
4.3	$N = 5$ の時のタイプB(遺伝子伸長型)における遺伝子重複法	68
4.4	$N = 5$ の時のタイプC(遺伝子結合型)における遺伝子重複法	69
4.5	$N = 5$ の時のタイプD(遺伝子拡張結合型)における遺伝子重複法	69
4.6	5次元Spにおける移民レートに対する ENES(PfGA ベース)	71
4.7	5次元Spにおける移民数に対する ENES(SSGA ベース)	71
4.8	10次元Spにおける移民レートに対する ENES(PfGA ベース)	72
4.9	10次元Spにおける移民数に対する ENES(SSGA ベース)	72
4.10	5次元Rsにおける移民レートに対する ENES(PfGA ベース)	74

4.11	10次元 R_s における移民レートに対する ENES(PfGA ベース)	74
4.12	R_s に対する ENES の次元依存性 (PfGA ベース)	75
4.13	R_s に対する ENES の次元依存性 (SSGA ベース)	75
4.14	5次元 M_i における移民レートに対する成功率 (PfGA ベース)	76
4.15	5次元 M_i における移民レートに対する ENES(PfGA ベース)	76
4.16	10次元 M_i における移民レートに対する成功率 (PfGA ベース)	77
4.17	10次元 M_i における移民レートに対する ENES(PfGA ベース)	77
4.18	5次元 R_d における移民レートに対する成功率 (PfGA ベース)	78
4.19	5次元 R_d における移民レートに対する ENES(PfGA ベース)	78
4.20	5次元 G_r における移民レートに対する成功率 (PfGA ベース)	80
4.21	5次元 G_r における移民数に対する成功率 (SSGA ベース)	80
4.22	5次元 L_a における移民数に対する成功率 (SSGA ベース)	81
4.23	5次元 L_a における移民数に対する ENES(SSGA ベース)	81
5.1	1次元強磁性イジングモデルに対するエネルギーの緩和過程. 縦軸は基底エネルギーとの差, 横軸は更新回数, サイズは $N = 50$, および $N = 100$, 初期温度は $T_0 = 1$, 温度スケジュールは $T(t) = T_0 \times 0.9^t$	89
5.2	平均場近似 (MFA) を用いた量子モンテカルロ法 (QMC) によるエネルギーの緩和. パラメータは, $N = 100$, $J = 1$, $T(t) = T_0 \times 0.9^t$, $T_0 = 1$, $H_x(t) = H_{x_0} \times 0.9^t$, $N_{\text{sample}} = 1000$, $E_0 = -(N - 1)J$. H_{x_0} の値はそれぞれ, 0, 0.1, 1.0.	94
5.3	平均場近似 (MFA) を用いた量子モンテカルロ法 (QMC) によるエネルギーの緩和. パラメータは, $N = 100$, $J = 1$, $T(t) = T_0 \times 0.9^t$, $H_x(t) = H_{x_0} \times 0.9^t$, $H_{x_0} = 1$, $N_{\text{sample}} = 1000$, $E_0 = -(N - 1)J$. T_0 の値はそれぞれ, 0, 1.0.	95
5.4	平均場近似 (MFA) を用いた量子モンテカルロ法 (QMC) によるエネルギーの緩和. パラメータは, $J = 1$, $T_0 = 0$, $H_x(t) = H_{x_0} \times 0.9^t$, $H_{x_0} = 1$, $E_0 = -(N - 1)J$. サイズ N はそれぞれ, 50, 100.	96
5.5	温度揺らぎによる分布と量子揺らぎによる分布	98
6.1	突然変異率 $p_m = 1/n$ によって摂動を受けた個体の分布 ($n = 24$)	105
6.2	$H = 8$ の時の子個体の分布	106
6.3	$H = 16$ の時の子個体の分布	106
6.4	評価回数の分布 (5-d Sphere model, PfGA)	107
6.5	評価回数の分布 (5-d Langerman's function, PfGA)	108
6.6	評価回数の分布 (5-d Langerman's function, SSGA)	108

第1章 序論

1.1 研究の背景

最適化手法には、大きく分けて厳密解法と近似解法がある。厳密解法とは、全ての解候補を列挙して最適解を厳密に求める方法である。解候補の数が有限で、列挙するのに実際上それほど記憶容量と計算時間を消費しなければ、この方法で十分である。しかし、解候補の数が大きく、列挙することが不可能な場合は、近似的な解を求める方法が採られる。これが近似解法(探索的手法)であり、組み合わせ最適化問題のサイズが大きい場合や、連続変数で定義される関数の最大最小値問題に対する解法がこれに当たる。

近似解を探索する手法にも、大別して確定的探索と確率的探索がある。確定的探索には、最急降下法や Brent 法 [1] などがあり、探索の方向がアルゴリズムによって一意的に決まっているのに対し、モンテカルロ法や進化的計算法などの確率的探索では、確率的に探索方向を決定する。確定的探索は局所的な探索に向いており、計算時間を重視し、できるだけ速く解を見つけたい場合に用いるが、確率的探索は大域的探索に向いており、できるだけ広い範囲を探索し、質の良い解を見つけることを目的とする。

モンテカルロ法は本来、力学系における緩和シミュレーションの方法であった。緩和とは、エネルギーの高い状態から低い状態への遷移である。このシミュレーション方法を最適化手法として採り入れたものがモンテカルロ法であり、統計力学をベースとしているため、統計力学的計算法とも呼ばれる。モンテカルロ法の基本的な考え方は、各要素の状態を局所的なルールに従って確率的に変更していくというものであり、この手法は、ニューラルネットワークなどのような工学的応用を始めとして、様々な分野で重要な役割を持つに至っている。

また、進化的計算法には、遺伝的アルゴリズムの他に、進化的戦略、進化的プログラミング、遺伝的プログラミングなどがあり、これらはいずれも生物の進化を模倣した計算モデルである。遺伝的アルゴリズムを最適化問題に適用する場合、問題の解候補を記号列からなる個体に対応させ、交叉、突然変異、および選択等の演算を繰り返し適用することにより適応度の高い個体を逐次生成し、近似解を取り出すという手順を採る。遺伝的アルゴリズムは関数最適化、組み合わせ最適化問題、機械におけるパラメータの最適設計などの多くの分野に応用されてきた。

1.2 研究の動機

探索アルゴリズムは解を探索することが目的である。では、優れた探索アルゴリズムとはどのようなものなのであろうか。優れた探索アルゴリズムは速く解を見つけるであろう。

あるいは、得られる解の精度が良い場合も優れた探索アルゴリズムであると言える。確率的に探索を行うわけであるから、解の精度が良いと言うのは、与えられた時間内に必ず近似解を見つけられることと等価である。探索速度と解の精度の問題は互いに矛盾した目的である。なぜなら、問題が非常に多くの極値を持つ場合、探索速度を上げるような設計を施すことによって局所解に落ちてしまい、なかなかそこから抜け出せずに、探索が失敗に終るといった状況が起こり得るからである。言い替えると、優れた探索アルゴリズムを開発する場合、収束性能を重視して局所探索を行う場合と、探索が成功することを重視して大域探索を行う場合とで考え方が異なり、従ってアルゴリズムの設計方針も異なってくるのである。

また次のような見方もできる。優れたアルゴリズムは単純であり、パラメータの種類が最小限である、あるいは、どのような問題にも適用できる汎用的なアルゴリズムも優れたアルゴリズムである。目的関数が明示的に与えられない場合にはより汎用的であることを重視する方が望ましく、従ってより多くの問題に対して優れた性能を発揮するようなアルゴリズムを考案することも重要であると考えられる。しかしながら、汎用的なアルゴリズムほどパラメータの数が多く、問題ごとに適切に設定しなければならないものである。

これらの視点を基に、確率的探索アルゴリズムである遺伝的アルゴリズムの特徴について、長所も短所も含めて列挙すると次のようになる。

1. 収束が遅い
2. 初期値に依存しにくい
3. 目的関数の特徴が未知であっても適用できる
4. パラメータが多い

これらは、例えば最急降下法などのような局所探索アルゴリズムなどと比較した場合の特徴である。第1点目は局所探索よりも大域探索を得意とすることを意味しているが、設計次第では局所探索を重点的に行うこともできる。しかしながら局所探索を重視する場合は、初期値に依存するアルゴリズムとなってしまうため、収束を速めることと初期値に依存しないアルゴリズムを目指すことは互いに相反する目標であることが分かる。また、3点目と4点目に関しても、どのような関数に対しても探索を行うことができる半面、そのために非常に多くのパラメータを適切に設定しなければならないのである。しかしながら、どのような関数に対しても適用できるということは注目すべき点である。そこで、遺伝的アルゴリズムがどのクラスの問題でどの程度実用に耐えるかという具体的な性能評価を行うことも重要となる。

一方、確率的探索アルゴリズムの中でも、モンテカルロ法や、その拡張であるシミュレーテッドアニーリング法などのような計算法は、局所探索と大域探索の中間に位置するとされ、遺伝的アルゴリズムと組み合わせて用いることで、大域探索と局所探索の両方を視野に入れたアルゴリズムの研究も行われている。モンテカルロ法自体は、大雑把に言えば進化的計算法の1種である。すなわち、個体数1の遺伝的アルゴリズムに相当していて、突然変異のみで状態更新していくアルゴリズムであるとみなせる。しかしながら、モンテカルロ法がどの程度の探索能力を持っているかは定かではないため、遺伝的アルゴリズムと組み合わせて用いることによりどれ程改善されるかという評価は断片的である。これについては、

モンテカルロ法を単体で用い、遺伝的アルゴリズムと比較することによって総合的に評価できると考えられる。

遺伝的アルゴリズムのパラメータ設計に必要な労力を、理論的な解析によって軽減しようとする試みもあるが、理論的にはごく単純化したモデルで、また、非常に簡単な目的関数に対してのみ成功しているだけであり、一般的な非線形関数に対しての解析は困難である。このような困難さの背景には、解を探索する個体集団の存在がある。個体の集団的な振舞いについては、分布という形でしか掌握できないためである。これは、物理で言う多体問題に相当しており、しかも個体数が可変であり、非平衡系を取り扱っていることに等しい程度の困難さを持っているからである。よって遺伝的アルゴリズムが優れていることの証明や、あるいは最適なパラメータを求めるための理論的アプローチには、どのような切り口が必要かを考えなければならない。

1.3 研究の目的

本研究では一貫して、優れた探索アルゴリズムを開発することを目的とする。優れたアルゴリズムを発見するための手段としては、実験的に試行錯誤を重ねながら発見していく方法と、理論解析によって最適な機構やパラメータを発見していく方法がある。前者に対しては、さまざまな非線形関数をベンチマーク問題として用い、系統的に調べ上げるという方法を採用することにする。また、優れたアルゴリズムであるならば、そのアルゴリズムの何が何に比べて優れているのかという点に着目しなければならない。すなわち、探索アルゴリズムの収束性能、探索能力、ロバスト性や設計の容易さなどが、その他のアルゴリズムや、従来手法などと比較することによって統合的に評価することができると考えられる。後者に対しては、目的関数を直接設定せず、個々の演算子がどのような振る舞いを示すかを調べることにより、最適なパラメータの設計指針を示すことができると考えられる。

1.3.1 汎用的なアルゴリズムの考案

遺伝的アルゴリズムには多くのパラメータが存在するが、解くべき問題に対して適切なパラメータを選ぶことによって、優れた探索性能を発揮する。しかしながら、解くべき問題の詳細な情報が得られない場合には、できるだけ汎用的なアルゴリズムを用いる方が望ましい。ここでは、どのような問題にも適用でき、ある程度の性能を発揮するような遺伝的アルゴリズムを考案する。このためには、パラメータの設定をなくし、アルゴリズムの内部状態が問題に対して適応的に変化するような機構を持たせることがポイントである。このアルゴリズムをパラメータフリー遺伝的アルゴリズムと呼ぶ。パラメータを全く用いないアルゴリズムは存在しないが、パラメータの設定が不要であるアルゴリズムは構築できると考えられる。これは設計者にとっても、利用者にとっても非常に便利なアルゴリズムとなることが期待される。

1.3.2 探索を速くするための工夫

探索を速くするためには、様々な方法が考えられているが、局所探索を重視するために局所解に陥ってしまうという弊害がある。そこで、遺伝的アルゴリズムの大域探索能力をそのまま引き継ぎ、なおかつ探索速度の速いアルゴリズムを考案することを目的とする。ここでは、遺伝的アルゴリズムの並列分散処理について考える。一般に並列処理は処理の高速化が目的であるが、並列の効果が、特に探索問題で顕著に現れることから、これを遺伝的アルゴリズムに実装した場合に、どの程度の収束性能の改善がなされるかという点を、4つの異なる実装法に分けて系統的に調べる。

1.3.3 探索を成功させるための工夫

与えられた時間内に探索を必ず成功させる、あるいは成功する確率を向上させる試みは重要である。これは同時に、解の質を向上させることにもつながる。この改良によって初期値に依存しない優れたアルゴリズムとなることが期待される。成功率を向上させるためのアイデアとしては、遺伝的アルゴリズムにおける個体集団の多様性の増加が最も妥当な方法であるが、単に個体数を増加させるだけならば、これはパラメータの調節の問題となってしまう。また、個体数を増加することによって個体の評価回数も増加してしまい、時間内に探索できなくなる可能性もある。そこで、優秀なスキーマ、すなわち遺伝子の部分配列を保存することによって多様性を増加させる方法を考える。本研究では、遺伝子重複説に基づいた遺伝的アルゴリズムを考案し、4種類の実装方法で実験を行い、遺伝子重複の有効性を探る。

1.3.4 統計力学的計算法と進化的計算との比較

モンテカルロ法などのような統計力学的計算法は、個体数が1個の進化的計算法に相当していると考えられる。そこで、モンテカルロ法の性能がどの程度優れているかについて、種々の関数最適化問題に適用し、遺伝的アルゴリズムとの比較を行う。この比較によって、確率的探索アルゴリズムの総合的な評価が可能となることが期待できる。

1.4 本論文の構成

本論文の構成は、次節で遺伝的アルゴリズムの基本概念と、従来法である、単純遺伝的アルゴリズムや、比較的優れた性能を持つ定常状態遺伝的アルゴリズムについてまとめ、従来法の特徴と問題点について述べる。

第2章では、どのような問題に対しても適用可能な汎用的なアルゴリズムとして、パラメータの設定をする必要のないアルゴリズム(パラメータフリー遺伝的アルゴリズム)を提案する。従来法と異なる点は、選択演算子の変則的で、結果的に個体数が可変となることである。これは、問題に応じて、また探索状況に応じて適応的に変化する機構を備えたアルゴリズムである。このアルゴリズムを、従来法と比較するため、国際コンテストで採用された

5関数と、それらを拡張した関数の合計9関数をベンチマーク問題として用いて実験を行い、収束性能、探索の成功率などの統計量を求める。

第3章では、収束性能を向上させるための工夫として、パラメータフリー遺伝的アルゴリズムの並列分散処理を提案する。基本的な考え方としては、遺伝的アルゴリズムにおける局所集団を複数用意し、これを並列に動作させるものである。また、局所集団間では個体の移民を行わせることにより、より速く解を見つける可能性がある。並列処理アーキテクチャには本来、一様分散型とマスタースレーブ型があり、また移民のさせ方にも、直接移民型と階層移民型の2種類を考慮することで、合計4種類の実装方法で並列化し、実験を行う。計算諸量は収束性能、成功率の統計量である。

第4章では、探索における成功率を向上させるための工夫として、遺伝的アルゴリズムに遺伝子重複を導入する。実装方法は、遺伝子の伸長、結合などを考慮した4種類を提案することにより、より優れたスキーマによる多様性の増大が見込まれる。実験は、ベースとなるアルゴリズムに定常状態遺伝的アルゴリズムと、パラメータフリー遺伝的アルゴリズムの2つを用い、4種類の実装法によって、合計8種類のアルゴリズムを構築し、実験を行う。計算諸量は収束性能、成功率の統計量である。

第5章では、進化的計算法以外の、他の探索アルゴリズムであるモンテカルロ法について述べる。モンテカルロ法はスピン系におけるエネルギーの最小値を求めるための手法として知られている。その拡張であるシミュレーテッドアニーリングは、温度揺らぎを徐々に小さくしていく方法であり、探索初期には大域的探索、探索後期には局所探索といった具合に、探索状況に応じた詳細な設計が可能となるため、比較的良好に用いられる手法である。ここではさらに、温度揺らぎを量子揺らぎに変更した計算法である、量子モンテカルロ法も用いる。量子モンテカルロ法自体は新しい手法ではないが、量子揺らぎを徐々に小さくしていくことを前提とすれば、より簡単な局所的ルールが得られる。また、問題をスピン系のエネルギー最小化問題に限定せず、他の関数に対しても適用できるよう、メトロポリス法を援用し、モンテカルロ法、量子モンテカルロ法と遺伝的アルゴリズムとの比較を行う。

第6章では、考察として遺伝的アルゴリズムの理論解析について述べる。ただし、理論解析自体は、ある目的関数を想定しなければならず、これにより解析は非常に困難となる。ここでは、遺伝的演算子のうち、突然変異と交叉について述べる。突然変異によって摂動を受けた個体がどの程度移動するかを、定量的な解析と実験により検証する。また、交叉については、親2個体の座標を固定した場合の、子個体の発生場所を統計的に求め、交叉が本質的に重要であることを示す。次に、本研究で主に用いる2つの評価基準(成功率と平均評価回数)に対する考察を行い、これらがある分布関数を介して密接に関係し合っている点と、これらが確率的探索アルゴリズム全般に共通する評価基準であることを示す。また、第3章の並列処理による収束性能の向上についても、これらの評価基準を用いて統一的に評価できることを述べる。

第7章は結論であり、本研究のまとめと今後の課題について述べる。

1.5 従来法としての遺伝的アルゴリズム

1.5.1 背景と概要

遺伝的アルゴリズム (GA) を初めとする進化的計算法 (EC) は, 生物進化のメカニズムを模倣した計算法であり, 工学的にも幅広く応用されている. GA は主に最適化手法として注目されているが, 歴史的には, 1859年, Charles Darwin が生物の遺伝と進化について「種の起源」を著し [2], それからヒントを得て, John Holland が1960年代に提案したのがGAである [3]. その後, 1970年代に入って De Jong により関数最適化問題を対象とした計算機実験が試みられた [4]. 彼はその後のGA研究の一つの指標として標準関数を導入した. また1989年にはGoldbergによってアルゴリズムの枠組が整理された [5]. 最近, 遺伝的アルゴリズムは他のスキームと発展的に統合され, Evolutionary Computation (EC) という分野を形成している [6].

GAの基本要素は, 個体集団 (population), 個体の適応度 (fitness value), および個体に対する演算 (genetic operation) の3つである. Goldbergによるアルゴリズムの枠組を簡単に説明すると, 次の通りである.

1. 初期集団の生成
2. 個体集団の適応度の評価
3. 個体集団に対する遺伝的演算

個体が持つ情報は, GTYPE と PTYPE の二層構造からなる. GTYPE (genotype) は遺伝子コードともいい, 細胞内の染色体に相当する. GA では, GTYPE が遺伝的操作の対象となる. また, PTYPE (phenotype) は表現型 (発現型) であり, 個体の大域的な行動や構造の発現を表す. GA では, PTYPE が個体の適応度を表す. 上記の枠組において, まず (1) は初期化であり, 一般的には, ある決まった数の個体をランダムに生成する. これにより, 探索空間に対して一様に分布するような個体集団が生成される. (2) の適応度の評価においては, 解くべき問題に応じて設計者が明示的に与えるものである. (3) の遺伝的オペレータには, 選択, 交叉, 突然変異などがある. 選択は, 基本的には, (2) で求めた適応度の高い個体が集団中に残り, 適応度の低い個体は淘汰されるような演算である. 交叉は, 複数の親個体 (通常は2個体) から部分遺伝子を取り出し, それらを組み合わせて新たな個体を生成する演算である. また, 突然変異は, ある確率で個体の遺伝子の一部を対立遺伝子に変更する操作である. これら一連の演算を行うことにより, 次世代の個体集団が生成されたことになる. この新たな集団に対し, (2) および (3) を繰り返し適用することにより, 個体集団は所望の解へと近づいていく. 最後に, 設計者が与えた終了条件を満たしていれば計算を終了する. 以上がアルゴリズムの大まかな流れである.

GA を最適化手法として用いるには, 以下の点に注意しなければならない.

- GTYPE から PTYPE への変換方法 (coding)
- 遺伝的オペレータの種類と適用方法
- 適応度の計算と選択方法

これらは、さまざまな方式が提案されているが、以下では、主に関数最適化問題で採用されている方式を重点的に説明する。

1.5.2 コーディング

GA で関数最適化問題を解く場合、問題を遺伝子で表現しておかなければならない。これを coding(符号化) と言う。また、この逆を decode(復号) と言う。実数変数で定義される関数の最適化を行う場合、決定変数ベクトル \mathbf{x} を n 個の記号列 \mathbf{X} で表現し、これを n 個の遺伝子座からなる染色体とみなす。記号列に用いる記号は、解くべき問題にも依存するが、この場合は、最も簡単な 0 と 1 の 2 値を用いる。符号化された染色体 \mathbf{X} が個体の遺伝子型 (GTYPE) であり、染色体に対応する変数 \mathbf{x} の値が表現型 (PTYPE) である。

GTYPE, すなわち染色体は有限個の記号列で表されなければならないので、定義域を離散化 (量子化) しなければならない。最も簡単な coding 方法は、変数を 2 進数で表し、その 0 と 1 の 2 進数 bit 列をそのまま用いるもので、binary coding と呼ばれる。binary coding で表される n bit の 2 進数列 β_i ($i = 1, 2, \dots, n$) を実数 x に変換する方法は、

$$x = L_1 + \frac{L_2 - L_1}{2^n - 1} \sum_{i=1}^n \beta_i 2^{n-i} \quad (1.1)$$

となる。ただし L_1, L_2 はそれぞれ定義域の下限と上限である。またこの x は離散化されているので、任意の x を binary coding に変換することはできないことに注意しなければならない。

本論文では、Gray coding と呼ばれる coding 方法を用いる。この方法は、binary coding と同様、0 と 1 の 2 値を用いるが、離散化された点のうち、隣り合う 2 点での bit 列が 1 bit のみ異なるような coding 方法である。ただし逆は成り立たない。すなわち 1 bit のみ異なる bit 列であっても、実数に変換して必ず隣り合うわけではないことに注意しなければならない。表 1.1 に、3 bit の場合の binary coding と Gray coding との対応を示す。binary coding から Gray coding への変換は、

$$\gamma_k = \begin{cases} \beta_1 & k = 1 \\ \beta_k \oplus \beta_{k-1} & k > 1 \end{cases} \quad (1.2)$$

で表される。ここで、 γ_k , および β_k はそれぞれ k bit 目の Gray coding bit, および binary coding bit である。また、 \oplus は排他的論理和 (XOR) を表す。逆に Gray coding から binary coding への変換は、

$$\beta_k = \left(\sum_{i=1}^k \gamma_i \right) \pmod{2} \quad (1.3)$$

となる。Gray coding を採用する理由は、反転 bit 数が少ない程、実数へ変換したときの値の差が少なく、すむ確率が高く、結果的に局所探索に向いているからである。例えば表 1.1 において、 $x = 5$ の時の Gray coding は [111] であるが、1 bit 反転すると [011], [101], [110] となり、これに対応する x の値はそれぞれ 2, 6, 4 である。従って近傍の値は 2 つ含まれる。これに対し binary coding では $x = 5$ の時、[101] を 1 bit 反転して [001], [111], [100] となり、それぞれ $x = 1, 7, 4$ である。この場合の近傍は 1 つのみである。その他の例題についても同様のことが示される。

表 1.1: binary coding と Gray coding との対応

x	binary coding	Gray coding
0	0 0 0	0 0 0
1	0 0 1	0 0 1
2	0 1 0	0 1 1
3	0 1 1	0 1 0
4	1 0 0	1 1 0
5	1 0 1	1 1 1
6	1 1 0	1 0 1
7	1 1 1	1 0 0

1.5.3 遺伝的オペレータ

選択, 淘汰

進化論的に考えると, 優秀な個体ほど交叉によって優秀な子個体を生成する確率が高い. 遺伝的アルゴリズムにおいて, 選択, 淘汰の目的は, 進化の妨げとなる劣悪な個体を排除し, 優秀な個体をより多く残すことである. つまり適応度の高い個体に, 子個体生成の機会を多く与えることによって, 集団全体の適応度をより高くするのである.

選択方法はこれまで, ルーレット方式, ランク方式, トーナメント方式など, 数多く考案されてきた. ルーレット方式は, 適応度に比例した割合で選択する方法である. まず架空のルーレットを用意し, 全ての個体にそれぞれ自分の適応度に比例した領域をもつルーレットのマス割り当てる. そしてランダムに回したルーレットの止まった位置にある個体を選択される. この操作を集団数が揃うまで繰り返し, 選ばれた個体群を新たな集団とするのである. 適応度 f_i をもつある個体 i がこの選択方法によって選ばれる確率 P_i は,

$$P_i = \frac{f_i}{\sum_j f_j} \quad (1.4)$$

となる. ただし適応度は, 実際の関数値をスケールし, 正の値に変更したものを用いる. スケールに関していくつかの技法が提案されており, 代表的なものに, 線形スケール, シグマスケール, ベキスケールなどがある.

ランク方式は, 適応度によって集団中の各個体にランクをつけ, 各順位に対してどの様な確率で子孫を残せるか予め決めておく. そして個体のランクに応じて, その個体の選択確率を決定するものである. この方法では, ランクの決定にソートが必要なことや, ランクに対する変換関数の選び方, さらに少数点以下の割り当ての問題などが生じる.

トーナメント方式は, まず集団からある個体数(トーナメントサイズ)をランダムに選び出してその中で一番良いものを(トーナメント方式で)選択する. この操作を集団数が揃うまで繰り返すというものである. この方法ではトーナメントサイズをどのように決定するかという問題が生じるが, 最良個体を選ぶのにスケールを必要としないので計算量は少なくすむ.

以上が代表的な選択方法である。計算量、収束性などの観点から、どの方式も万能ということはなく、問題に応じて使い分けるべきであるが、トーナメント方式が相対的に優れているという報告もある。

交叉

交叉 (crossover) は、2つの親個体の遺伝子を組み替えて子個体を生成する操作である。交叉点の数から、1点交叉、複数点交叉、一様交叉に分けられる。1点交叉は、遺伝子上のある1点を交叉点に選び、その交叉点を境にどちらの親の遺伝子型を受け継ぐかを決定する方法である。複数点交叉は、例えば n 点交叉はその交叉点が n 点あり、遺伝子を交互に受け継ぐ方法である。一様交叉は、複数点交叉の n をランダムに選ぶものに等しい。一般に、1点交叉よりも、複数点交叉や一様交叉のほうが優れているとされている。

2つの親個体のハミング距離が大きく、すなわち遺伝子型が全く異なるような場合は、交叉によって生じる2つの子個体も親とのハミング距離は大きくなり、見かけ上全くばらばらとなる。この多様性によって、結果的に大域的探索を行うことができるのである。集団ベースのGAでは、探索初期でこのような状況が頻繁に現れる場合がある。また、探索後期においてよく見られる、2つの親個体が非常に似通った遺伝子型を持つ場合は、子個体も親個体に似てしまい、結果的に局所探索を行う割合が高くなるのである。これらの解釈は、集団ベースのGAにおいて、交叉が重要な役割を果たしていることの原因でもある。しかしながら、似通った個体が増えるということは、同時に個体の大半が同じ遺伝子を持つようになる事を意味し、このため、それ以上探索が進まなくなるという初期収束の問題もあり、これを解決するため、集団のなかに似たものをあまり増やさないような工夫も研究されている。

突然変異

突然変異は、遺伝子を一定の確率 (突然変異率) で変化させる演算である。突然変異率が大きい場合、後述する部分順序列であるスキーマが破壊され、ランダム探索に近くなってしまう。逆に突然変異がない場合は、交叉による初期集団の遺伝子の組合せ以外の解空間を探索することはできないため、特に局所探索ができなくなる。また、交叉のみの探索では、集団中で全ての個体が同じ遺伝子をもつ初期収束が起こる可能性もある。

突然変異率については、問題が単峰性関数のときは遺伝子長の逆数とするのが良いとされている。これは世代ごとに各遺伝子につき平均一つの遺伝子座で変異が起こることを意味する。また初めのうちは大きめの変異率を採用し、世代を重ねるにつれて小さくしていく方法も研究されている。これは、モンテカルロ法におけるシミュレーテッドアニーリングに似た計算手法である。しかしながら、一般の多峰性関数では最適な突然変異率は定まらないので、探索状況に応じて変化させる方法も考えられている。

1.5.4 スキーマ定理

スキーマとは個体集合中の、特定の遺伝子型についてのテンプレートにマッチする部分集合を指す。スキーマによって集団の時間発展をとらえると、GAの興味深い解析が可能と

なる。

遺伝子記号 $\{0, 1\}$ の他に, ワイルドカード $*$ を用いると, 個体集合のスキーマは $\{0, 1, *\}$ のいずれかの値をとる記号列で定義される. 例えば, スキーマ $H = \{10*1*\}$ は, 4 個体からなる集合 $\{10010, 10011, 10110, 10111\}$ を表す.

次に, スキーマ H に対して, そのオーダ $O(H)$ を H 中のワイルドカード記号以外の記号の数で定義する. 例えば, $O(10*1*) = 3$ である. また, スキーマ H 中のワイルドカード記号以外で左 (右) 端にある記号を左 (右) 端定義記号と呼び, スキーマ H の定義長 (defining length) $\delta(H)$ を, 両端定義記号間の距離で定義する. 例えば, $\delta(10*1*) = 3$ である.

実際の GA の中で, スキーマがどのような振舞をするかを追ってみることにする. t 世代目の個体集合 $P(t)$ 中でスキーマ H を持つ個体数を $m(H, t)$ で表すと, まず複製演算子によって, スキーマ H に含まれる各個体 A_i は, 次世代に確率 $f(A_i) / \sum_i f(A_i)$ で複製されるから, $m(H, t+1)$ は,

$$m(H, t+1) = m(H, t) \frac{f(H, t)}{\bar{F}(t)} \quad (1.5)$$

となる. ここで $f(H, t)$ は, スキーマ H に t 世代で含まれる個体の平均適応度であり, $\bar{F}(t)$ は, t 世代の全個体の平均適応度である. これは, 全個体の平均適応度よりも高い適応度を持つスキーマは増加し, 逆に低いスキーマは減少する傾向にあることを示している.

次に, 交叉による変化を評価する. スキーマ H の遺伝子長が l で, 定義長を $\delta(H)$ とする. ランダムな位置で一点交叉が行われると, 両端定義記号の間に交叉点を選ばれるときにスキーマは破壊される. 交叉自体が行われる確率を p_c とすると, スキーマが H が保存される確率 p_s は,

$$p_s \geq 1 - p_c \frac{\delta(H)}{l-1} \quad (1.6)$$

となる.

最後に, 突然変異の確率を p_m とすると, ワイルドカード記号以外の遺伝子に突然変異が作用し, 遺伝子 1 つが反転しない確率が $1 - p_m$ であることから, スキーマ H が残る確率は,

$$(1 - p_m)^{O(H)} \sim 1 - p_m O(H) \quad (p_m \ll 1) \quad (1.7)$$

となる. 以上の関係式から, 結局次の近似式を得る.

$$\begin{aligned} m(H, t+1) &\geq m(H, t) \frac{f(H, t)}{\bar{F}(t)} \left[1 - p_c \frac{\delta(H)}{l-1} \right] [1 - p_m O(H)] \\ &\geq m(H, t) \frac{f(H, t)}{\bar{F}(t)} \left[1 - p_c \frac{\delta(H)}{l-1} - p_m O(H) \right] \end{aligned} \quad (1.8)$$

これはスキーマ定理と呼ばれるもので, GA における基本原理となっている. この式の意味は, 定義長が短く, オーダが小さく, 適応度が平均以上のスキーマの数は指数関数的に増大するというものである. このようなスキーマを Building Block と呼ぶが, GA ではこの Building Block をうまく組み合わせて最適値探索を行っているのである.

また, GA で実際に評価されるのは, 長さ L の遺伝子に対し, 2^L 個の文字列であるが, そのとき裏では 3^L 個のスキーマが間接的に, しかも同時に評価されていることになる. これを内在的並列性 (implicit parallelism) と呼ぶ. この並列性によって, GA は効率的に解を探索できると考えられる.

1.5.5 従来手法 1: 単純遺伝的アルゴリズム

遺伝的アルゴリズムの中で特に代表的で簡単な、単純遺伝的アルゴリズム (SGA) について説明する。これは、Goldberg[5] によって提案されたものである。SGA の構成は次の通りである。

1. 初期集団の生成: 一様乱数を用いてランダムに記号列を決め, M 個の個体を生成する.
2. 適応度の評価: 各個体の適応度を, 関数の値から求める.
3. 遺伝的演算:
 - (a) 選択: 世代 t における個体集合 $P(t)$ から, ルーレット方式を用いて, $P'(t)$ を生成する.
 - (b) 交叉: 個体集合 $P'(t)$ 内の個体をランダムにペアリングし, 交叉率 p_c で 1 点交叉を行う. これにより $P''(t)$ が生成される.
 - (c) 突然変異: $P''(t)$ 内の個体の各遺伝子座の遺伝子を, 突然変異率 p_m で反転させ, 次世代の個体集合 $P(t+1)$ とする.
4. 判定: 終了条件を満たせば計算終了. そうでなければ 2 に戻る.

以上のように, SGA を実現するためには, 次の 4 つのパラメータを設定しなければならない。

- T : 計算終了までの世代数
- M : 個体数
- p_c : 交叉率
- p_m : 突然変異率

これらをまとめて, (T, M, p_c, p_m) と表記する。

次に, SGA の具体的な構成例と実験を示す。次式は, De Jong の F2 と呼ばれるテスト関数 (Rosenbrock's saddle) である。

$$F_2 = 100(x_1^2 - x_2)^2 + (1 - x_1)^2 \quad (1.9)$$

$$(-2.048 \leq x_i \leq 2.048, \quad i = 1, 2)$$

この関数は, 最小値を求める場合と最大値を求める場合の 2 種類の使い方ががあるので注意を要する。最大値を求める場合は, いわゆるだまし問題の典型的な例であるが, ここでは最小値を求める。

計算のため, まず, 定義域に対する変数の精度を設定し, それに応じた長さの遺伝子を用意する。ここでは, 精度を $\varepsilon = 10^{-3}$ とし, 定義域の大きさを L とすると,

$$l_i = \log_2(L/\varepsilon) = 12 \quad (1.10)$$

となる. すなわち, 1変数当たり 12 bit である. また, 関数は2次元であるから, 合計 24 bit の長さの遺伝子となる. 個体の適応度を評価するために, この遺伝子が Gray coding で表されていると仮定して, これを binary coding に変換し, さらに実数に変換して関数の値を求める. この関数の値の符号を反転することにより, 最も簡単に適応度を求めることができるのであるが, ここでは, 集団中の最悪個体の関数の値との差で与え, これを規格化したものをルーレット方式のマス大きさに対応させる. すなわち, i 番目の個体が選択される確率は,

$$P_i = \frac{f_{\text{worst}} - f_i}{\sum_j (f_{\text{worst}} - f_j)} \quad (1.11)$$

となる. 1回目の選択では, M 個体から M 個体を選ぶので, 結局何もせず, 選択を行うのは2世代目からである. また, 交叉では, 交叉率が p_c であるから, $M \times p_c/2$ 回の交叉を行い, $M \times p_c$ 個の子個体を生成することになる. よって $P''(t)$ や $P(t+1)$ の個体数は $M + M \times p_c$ となっている. これにより, 2世代目からの選択は, $M + M \times p_c$ 個から M 個を選択することになる.

図 1.1 は, $(T, M, p_c, p_m) = (100, 100, 0.8, 0.001)$ をパラメータとして用いた場合の計算結果である. 遺伝演算子はルーレット方式による選択, 1点交叉, および bit 反転による突然変異である. 真の最小値は 0 であるので, 最良個体は第 40 世代以降は探索があまり進まず, また最良解を保存していない. しかしながら, 集団全体としては収束していることが分かる.

また, 図 1.2 は, この計算での探索過程として, 各個体の座標を示したものである. 初期状態では, 探索空間に一様に分布していた個体は, 探索が進むにつれ次第に解に近付き, 第 100 世代では, 個体のうちのほとんどが, 解 $x_1 = x_2 = 1$ 付近に集まってきていることが分かる.

注目すべき点として, 第 20 世代以降の個体の分布では, 個体が x_1 軸, または x_2 軸に沿って規則的に配列しようとする傾向がうかがえる. これは, 1点交叉を行うことによる影響である. 遺伝子は x_1 と x_2 をそれぞれ Gray coding したものを連結させている構造であり, 交叉点が x_1 側にあれば x_2 の部分はスキーマとして残り, 逆に x_2 側が交叉点であれば x_1 の部分はスキーマとして残るためである.

1.5.6 従来手法 2: 定常状態遺伝的アルゴリズム

次に, 遺伝的アルゴリズムの中でも比較的優れているとされている, 定常状態遺伝的アルゴリズム (SSGA)[7] について説明する. SSGA の構成は次の通りである.

1. 初期集団の生成: 一様乱数を用いてランダムに記号列を決め, M 個の個体を生成する.
2. 適応度の評価: 各個体の適応度を, 関数の値から求める.
3. 遺伝的演算:
 - (a) 交叉: 個体集合内の 2 個体をランダムに選択し, 一様交叉を行う. これにより 2 個の子個体が生成される.
 - (b) 突然変異: 2 個の子個体の各遺伝子座の遺伝子を, 突然変異率 p_m で反転させ, 個体集合に戻す. これにより個体数は $M + 2$ となる.

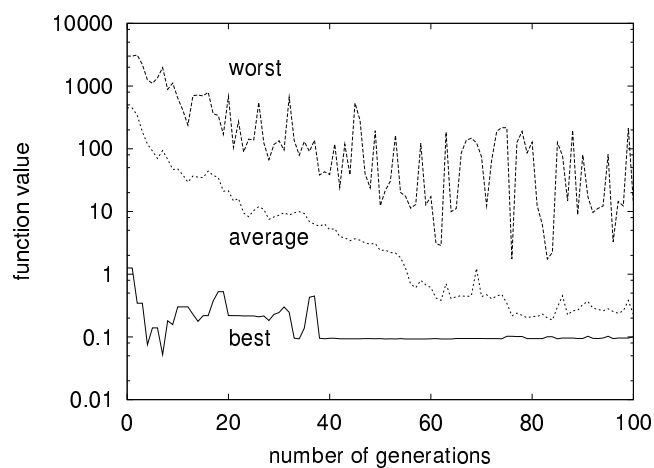


図 1.1: De Jong の F2 に対する個体の収束の様子 (SGA)

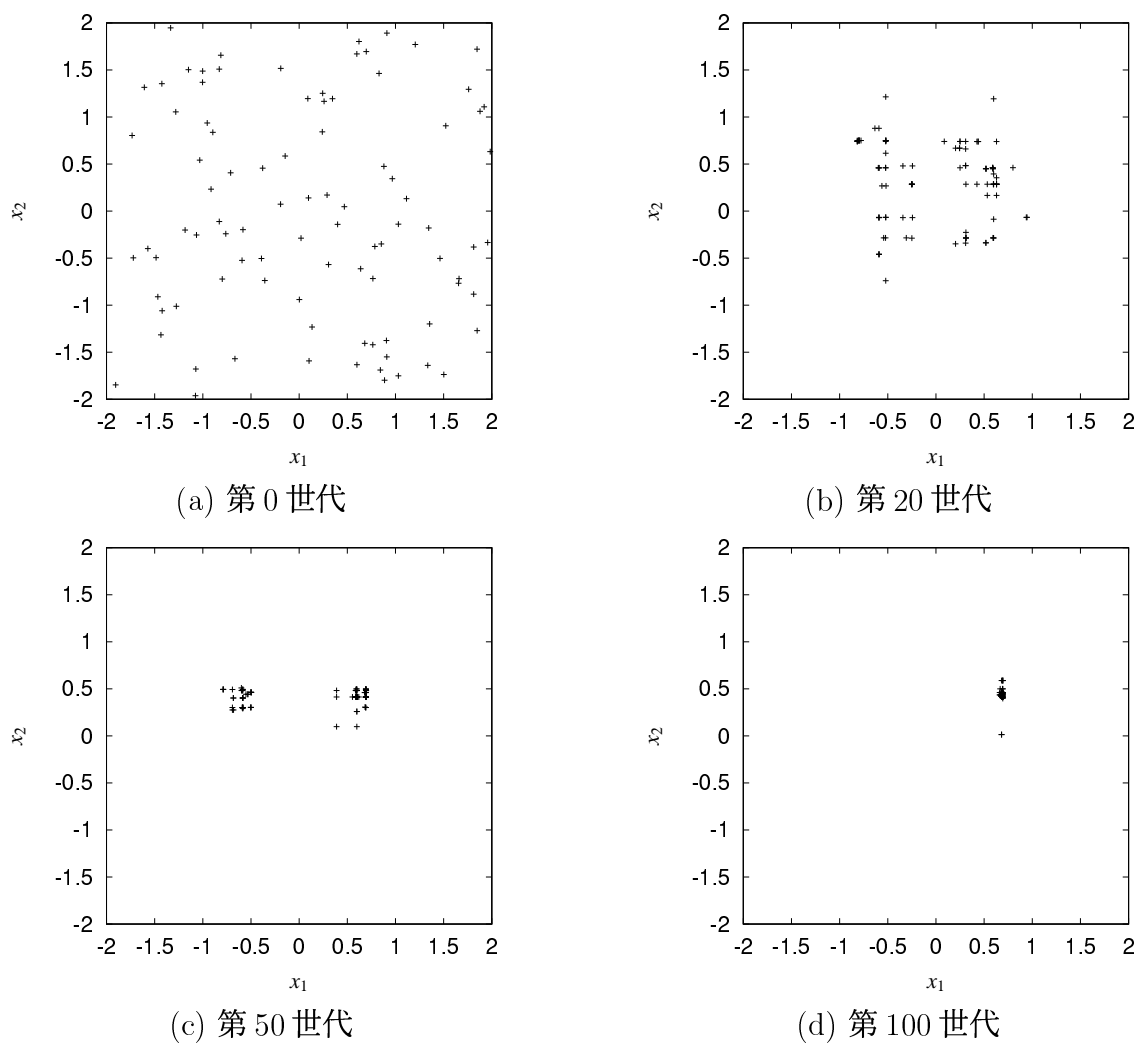


図 1.2: 探索過程における個体の分布 (SGA)

(c) 淘汰: $M + 2$ 個の個体集合から, 適応度の最も低い2個体を淘汰する.

4. 判定: 終了条件を満たせば計算終了. そうでなければ3に戻る.

ここで用いる淘汰の方法は, "delete least fit" 戦略 [7] と呼ばれる. SSGA のパラメータは,

- T : 計算終了までの評価回数
- M : 個体数
- p_m : 突然変異率

である. 交叉率は与えていないが, SGA との比較から, SSGA の交叉率は1であるとみなせる. また, 世代数ではなく評価回数を与えているのは, 第2章以降で取り扱うアルゴリズムが世代数を明確に定義できないためであり, よって共通した終了条件とするためである. もちろん SSGA 自体の世代数は定義でき, 評価回数との関係は, 世代数 = 評価回数 / 個体数である. 以上のパラメータをまとめて, (T, M, p_m) と表記する.

次に, SSGA を前述の De Jong の F2 に適用してみる. 図 1.3 に示す計算結果は, $(T, M, p_m) = (10000, 100, 0.05)$ をパラメータとして用いた場合である. 遺伝演算子は, 選択に最悪2個体を淘汰する delete least fit 戦略を用い, 一様交叉, bit 反転による突然変異を用いた. 図から分かるように, delete least fit 戦略では, 最良個体は必ず保存されるため, best で示される最良個体の関数値は減少し続ける. 100 回の評価を SGA の1世代と対応させることにより, SGA の場合に比べ, 最良個体の関数値や平均値はより速く収束していることが確認される.

また, 図 1.4 は, この計算での探索過程として, 各個体の座標を示したものである. 初期状態では, 探索空間に一様に分布していた個体は, 探索が進むにつれ次第に解に近付き, 第100世代では, 個体のうちのほとんどが, 解 $x_1 = x_2 = 1$ 付近に集まってきていることが分かる.

注目すべき点としては, 図 1.4 の (b) に見られるように, 個体集団が関数の「谷」に沿って分布している点である. この関数の「谷」とは, 曲線 $x_1^2 = x_2$ のまわりにある等高線に沿った部分である. このような探索過程を示す理由は, SSGA の選択演算子の性質が関与している. すなわち, 良い個体を積極的に選択するのではなく, 悪い個体を淘汰するだけの消極的な選択方法であるが, それがかえって個体集団を正しい方向へ漸近的に導くこととなるのである. これは例えば, 図の (b), (c), (d) のような状況においては, 親個体として選ばれる2個体のハミング距離は非常に近くなっていて, この親から生成される子もまた, 親からのハミング距離が近くなる確率が高くなるため, 局所探索によって良い子個体を生成する確率が高くなるという解釈もできるのである.

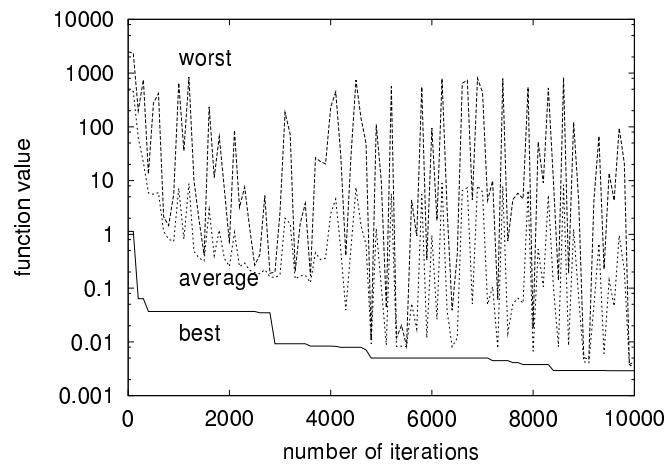


図 1.3: De Jong の F2 に対する個体の収束の様子 (SSGA)

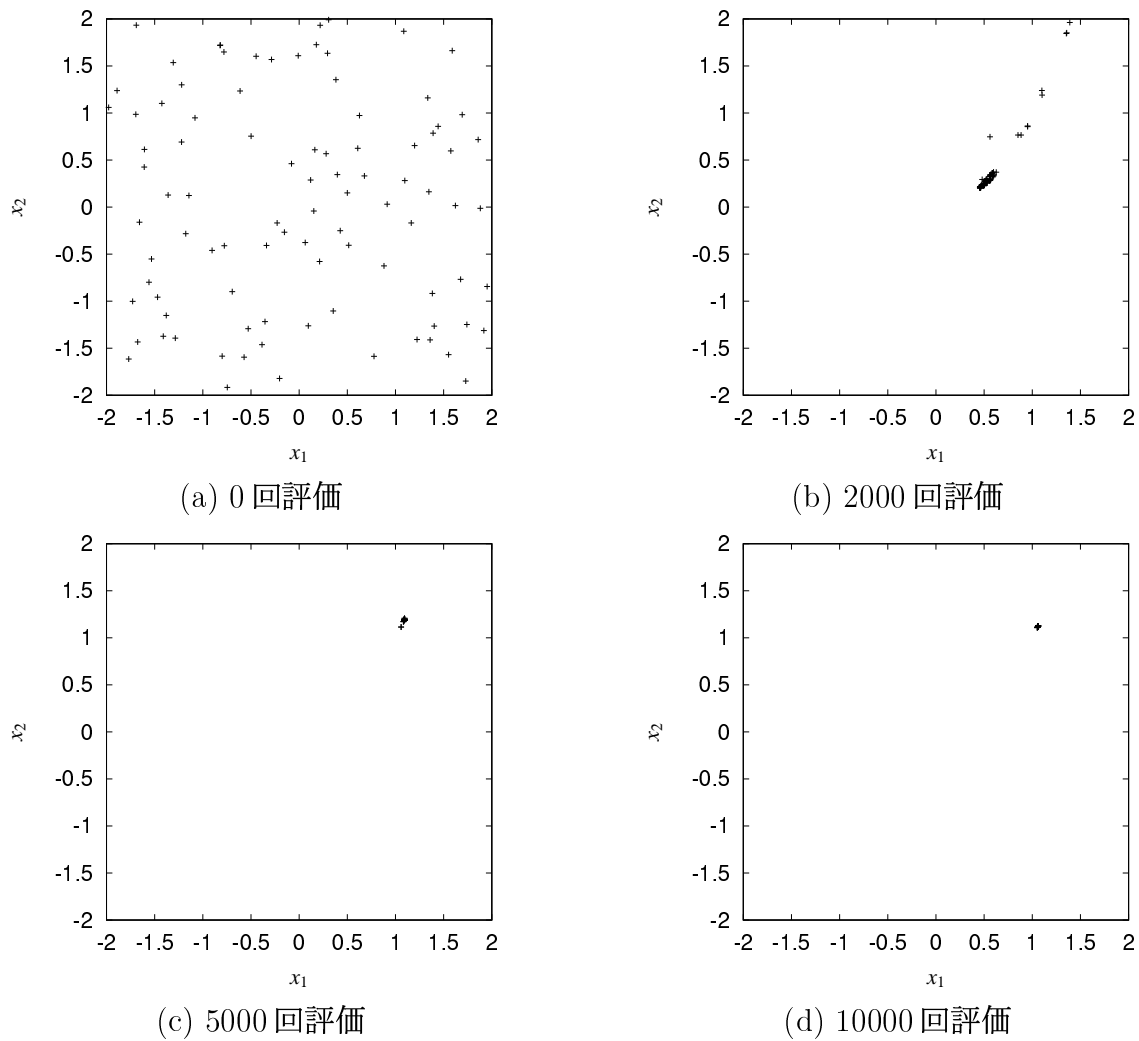


図 1.4: 探索過程における個体の分布 (SSGA)

第2章 パラメータフリー遺伝的アルゴリズムの提案と性能評価

2.1 概要

本章では、汎用的なアルゴリズムを構築する目的で、パラメータ設定を必要としないパラメータフリー遺伝的アルゴリズム (PfGA) を構築し、その性能を評価する。このアルゴリズムのベースとなる考え方は木津氏らによるものであり、著者らによって更に発展させたものである [12]。

GA を現実の問題に適用しようとした場合、パラメータ設定が設計者の重大な課題となる。そこで、何らかの方法で自動的に最適なパラメータを求めるか、あるいは必要最小限の定数パラメータのみを用いるという設計法などが研究されている。その中で、従来研究の1つにメタ GA というものがある。これは、通常の GA が最適解を探索するのに対して、その GA で用いるパラメータを別の GA で決定するものである。イメージとしては階層構造をもった GA であるとみなせる。しかしながら、少なくとも2つの GA を用いるので計算量が多く、アルゴリズム自体も複雑になってしまい、あまり好ましい方法ではない。

本章で提案する PfGA は、問題に応じて最適なパラメータをアルゴリズム自身が適応的に見つけ出す機構を備え、初期集団数、交叉率、突然変異率などの遺伝的パラメータの設定をする必要のないアルゴリズムである。パラメータのないアルゴリズムは存在しないが、パラメータ設定をしないですむようなアルゴリズムは存在するというのが、基本理念である。PfGA はどのような環境にも適応できるように、集団数を適応的に変化させる点が特徴である。

PfGA と関連した、動的に空間を探索する機構を備えた従来研究としては、GA の探索プロセスを forking しながら有望領域を探索するアルゴリズム Forking GA [13]、局所探索の範囲を可変にするもの [14]、SA (Simulated Annealing) を集団で並列に実行するもの [15] などがあるが、いずれも遺伝的パラメータの値は提案者が各個人の経験に基づいて設定している。

2.2 アルゴリズムの基本構造

PfGA [12] は、大域的探索と局所的探索のバランスを保つようなアルゴリズムである。言い替えると、柔軟性と安定性の両方を兼ね備えたシステムである。これを実現するために、次の2つの概念を採り入れる。

- 可変な集団数

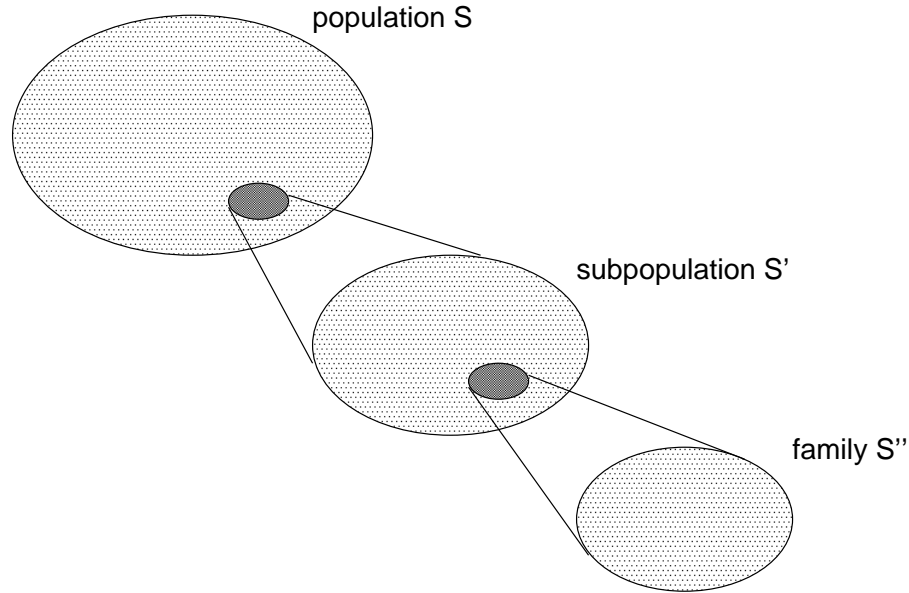


図 2.1: PfGA における集団の視点

- 非対称な突然変異

集団数を変化させるには、変則的な選択演算子を導入することにより達成される。これによって、個体数が多い場合と少ない場合で探索の仕方が変わると考えられる。また2つ目の非対称な突然変異の適用によって、大域的探索と局所的探索の両方を行うことができると考えられる。

PfGA ではパラメータを明示的に設定しないが、初期個体数などは設定しないとアルゴリズムそのものが成り立たなくなってしまう。よって初期個体数は1とする。大まかな流れは、まず全探索空間から1つの局所集団 (sub population) を抽出する。この局所集団を構成するメンバーが、ランダムに選んだ1つの初期個体である。さらにこの局所集団から、家族 (family) と呼ばれる個体集団を生成する (図 2.1)。交叉、突然変異、選択の3つの遺伝的演算はこの家族に対して行う。選択に関しては、家族内の個体の適応度の大小に応じて局所集団に残す個体数を変化させるようにする。以上の手続きをまとめると次のようになる。ただし、 S を全探索空間、 S' を局所的な副集団とする。

1. S から1個体を無作為に取り出し、これを S' の初期個体とする。
2. S から1個体を無作為に取り出し、これを S' に追加する。
3. S' から無作為に2個体選び出して、これを親 P_1 , P_2 とし、多点交叉を行う。
4. 交叉によって生成された2個体の子 C_1 , C_2 のうちの1個体に対し、突然変異を適用する。
5. P_1 , P_2 , C_1 , および C_2 の計4個体を、適応度の大小に応じて1個体から3個体を局所集団 S' に戻す。

6. 局所集団の個体数が $|S'| \geq 2$ なら 3 へ戻り, $|S'| = 1$ なら 2 に戻って繰り返す.

ただし便宜上, この 1 ループを PfGA における 1 世代と呼ぶことにするが, 個体数が可変であるため, 従来法のような世代の明確な定義ができないことに注意しなければならない. 次にこのアルゴリズムの詳細について説明する.

2.2.1 局所集団, および家族

PfGA では, 局所集団サイズは一定ではなく可変である. サイズの最小値は 1 である. アルゴリズムが保持する個体が局所集団であり, これらが探索に関わっている. また, 局所集団から家族を生成するのであるが, 局所集団サイズが 1 の時は, 家族を生成することはできない (すなわち交叉できない) ため, 全探索空間から新たな個体を発生させ, 局所集団に加えておく. このようにして, 局所集団数を 2 以上としておき, ここから親個体を任意に 2 つ選び, 家族を生成する.

遺伝的アルゴリズムの従来法と比べて個体数が少ないため, 一見すると局所探索を行っているようにも解釈できるが, 実際は上述のように, ある割合で全探索空間から新たな個体をランダムに発生させているため, ランダムサーチの側面も持ち合わせているのである. すなわち, 大域的探索と局所的探索のバランスをとっている探索法であるとも言える.

2.2.2 交叉, 突然変異

親として選ばれた 2 個体は多点交叉を行い, 子個体を 2 つ生成する. ただし交叉点の数と場所はランダムに決定する. 一点交叉の場合は第 1 章で述べたように, スキーマの保存に優れているが, その代わりに探索できる範囲が極端に狭められてしまうため, 大域的な探索には向いていない. よって多点交叉を行う.

また, 生じた子個体のうち 1 個体に対し, ランダムな確率で inverse mutation と呼ばれる突然変異を行う. これは, ランダムに選んだ 2 点間の遺伝子全てを対立遺伝子に変更するものである. ただし, 遺伝子の中間点と端の点を選ばれる確率を同等にするため, 遺伝子をリング状にしておき, 周期的にする. 具体的には, 長さ n の遺伝子 γ_i に対して, 一様整数乱数 n_1, n_2 を引き, $n_1 < n_2$ ならば n_1 から n_2 までの遺伝子を反転させる. すなわち,

$$\gamma_i \rightarrow 1 - \gamma_i, \quad (n_1 < i \leq n_2) \quad (2.1)$$

である. 逆に $n_2 < n_1$ ならば, n_2 から n_1 までの遺伝子はそのままで, それ以外を反転させる. すなわち,

$$\gamma_i \rightarrow 1 - \gamma_i, \quad (1 < i \leq n_2, \quad n_1 < i \leq n) \quad (2.2)$$

である. これを図示すると図 2.2 のようになる. mutation block が突然変異を適用する範囲である.

また, もう一方の子個体は突然変異は適用されない. このように, 生じた子個体を差別化し, 非対称性を持たせることにより, 突然変異を受けた子個体は柔軟性に富み, 大域的探索を行うことができる. また, 突然変異を受けなかった子個体は安定性を持ち, 探索を加速する役割を持つ.

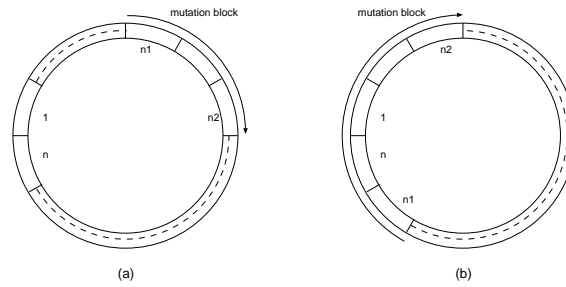


図 2.2: inverse mutation, (a) $n_1 < n_2$ のとき. (b) $n_1 > n_2$ のとき.

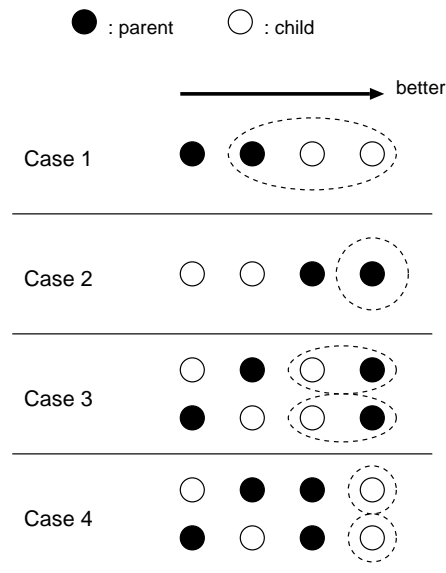


図 2.3: PfGA における選択則

2.2.3 選択, 淘汰

家族 4 個体に対し, 選択, 淘汰を行う. 図 2.3 は親子の適応度の高いものから順に並べたときを場合分けしたものであり, 破線で示される枠内が実際に選択される個体を表す. まとめると次のようになる. ただし, 親個体を P_1, P_2 , 子個体を C_1, C_2 とする.

- **Case 1:** C_1, C_2 共に P_1, P_2 よりも適応度が大きい場合. この場合は, C_1, C_2 は局所集団 S' に残り, さらに P_1 と P_2 の適応度の大きい方も S' に残る. この時 $|S'|$ は 1 増加する. このケースでは, 個体群は局所探索に向かっていると考えられる. よって個体数の増加は, 最適解への収束を加速させるための工夫である.
- **Case 2:** C_1, C_2 共に P_1, P_2 よりも適応度が小さい場合. この場合は, P_1 と P_2 の適応度の大きい方が S' に残り, $|S'|$ は 1 減少する. この場合は探索が失敗に終る典型的なケースであるため, 親 P_1, P_2 の組合せが適切でなかったと解釈し, 親の良い方のみを残すという方法を採用するのである.

- **Case 3:** C_1, C_2 の適応度の大きい方が, P_1, P_2 のうち適応度の大きい方よりも小さく, かつ, P_1, P_2 のうち適応度の小さい方よりも大きい場合. この場合は, C_1, C_2 の適応度の大きい方と, P_1, P_2 の適応度の大きい方を S' に残す. これにより $|S'|$ は変化しない. このケースでは, 悪い方の親を良い方の子で置き換えるのであるから, 最良解は更新されないが, 局所集団としては, わずかに解に近づくと考えられる.
- **Case 4:** C_1, C_2 の適応度の大きい方が, P_1, P_2 の適応度の大きい方よりも大きく, かつ, C_1, C_2 の適応度の小さい方が, 両親の適応度の大きい方よりも小さい場合. この場合は, C_1, C_2 の適応度の大きい方が S' に残る. これにより $|S'|$ は 1 減少する. このケースは Case 1 と同様, 探索が成功した場合であるため, 最良子個体を残すのは当然である. また, 個体数を減らすのは, このケースにおける探索状況が Case 1 よりも進みにくくなっているため, 初期収束を防ぐためにこのような方法を採用するのである.

通常の GA では, 大域的探索が必要なときは個体数を増加し, 局所探索が必要なときは個体数を減少させるのが普通であるが, 上記のケースでは, その逆となっているように見える. しかしながら, 個体群としてのダイナミックな探索状況を想定して設計しているため, このようになるのである.

また, いずれの場合も, 最良個体は必ず選択しているので, エリート保存戦略の一種であることが分かる. 個体の適応度に関しては, 最小値探索を行う場合は, 関数値の符号を逆にしたものを用いればよいのであるが, 必要な情報は 4 個体の適応度の大小関係のみである. このように, PfGA は初期集団数は 1, 交叉率も 1, 突然変異率はランダムという具合に, 見かけ上, 遺伝的パラメータを明示的に設定しておらず, 選択, 淘汰に関しても目的関数のスケールリングなども行わないシンプルなアルゴリズムである.

2.3 実験 1: De Jong の F4

PfGA の実行例として, 図 2.4, 2.5 に De Jong の F4 と呼ばれる関数に対する探索の様子を示す. この関数は次式で与えられる.

$$F_4 = \sum_{i=1}^{30} ix_i^4 + N(0, 1) \quad (2.3)$$

$$(-1.28 \leq x_i \leq 1.28)$$

ただし, $N(0, 1)$ は, 平均 0, 分散 1 のガウス雑音 (Gaussian noise) であり, 個体の適応度 (関数値) を計算する毎にガウス乱数として与えた. 関数値が動的に変化する関数の代表例である. 遺伝子長は 1 次元当たり 22 bit とし, 評価回数を 10,000 回とした.

図より, 最良個体の適応度は急速に 0 に近付き, 4,000 を越えてからは負の値をとるようになる. また, 個体数は 1 から 5 までをとり, 時間的に変化する環境に対して, 動的に適応していることが分かる. このように PfGA は, 家族内の個体の適応度の大小に応じた Case 1 ~ 4 の切替えによって, 局所集団 S' のサイズを適応的に変化させ, 大局的な探索と局所的な探索のバランスを取りながら進化するのである.

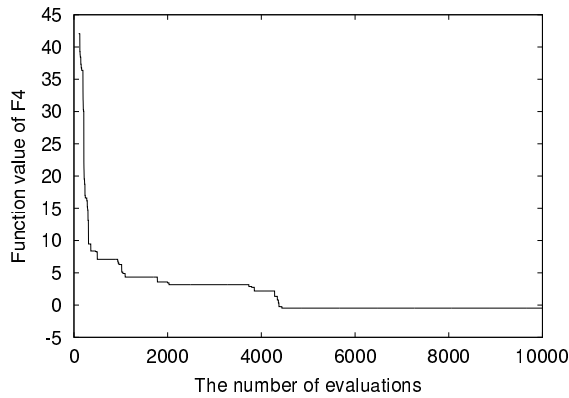


図 2.4: De Jong の F4 に対する PfGA の収束の様子

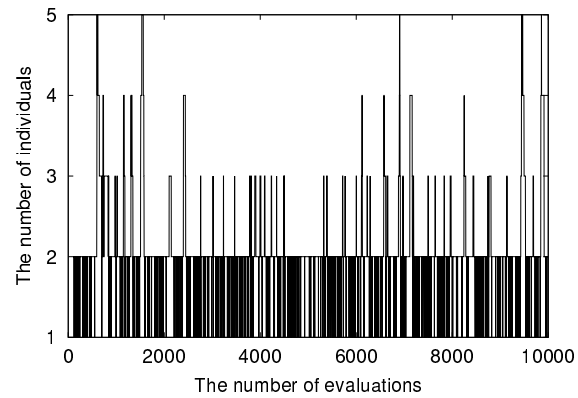


図 2.5: De Jong の F4 に対する PfGA の局所集団サイズの変化

2.4 実験 2: ベンチマーク問題

PfGA のパフォーマンスを他のアルゴリズムと比較するため、上下限付き多次元関数最適化 (最小値探索) 問題を用いて調べる. 以下にその 9 つの関数の関数型と、定義域、1 変数当たりの遺伝子長、VTR(Value To Reach) を示す. VTR は各関数の解の精度を測る指標である. これらの関数は、一般の関数型を構成するように、単峰性 (unimodal) か多峰性 (multi-modal) か、変数 x_i に対して対称 (symmetric) か非対称 (asymmetric) か、変数分離可能 (separable) か不可能 (inseparable) かの 8 つの場合に分類したものである.

$$F_{\text{Sp}} = \sum_{i=1}^N (x_i - 1)^2 \quad (2.4)$$

$$-5 \leq x_i \leq 5, (24\text{bit}), VTR = 10^{-6}$$

この関数は Sphere model (Sp) であり、De Jong の F1 とも呼ばれる. 極小値は 1 つのみであるので、単峰性関数である. 解は $x_i = 1, (i = 1, 2, \dots, N)$ の時、0 となる.

$$F_{\text{Ds}} = \sum_{i=1}^N \left(\sum_{j=1}^i x_j \right)^2 \quad (2.5)$$

$$-65.536 \leq x_i \leq 65.536, (27\text{bit}), VTR = 10^{-4}$$

この関数は Schwefel's Double Sum (Ds) と呼ばれる単峰性関数である. 解は $x_i = 0$ の時、0 となる. また、解付近の等高線は楕円であり、変数分離不可能な関数である.

$$F_{\text{Rs}} = \sum_{i=1}^N (x_i - r_i)^2 \quad (2.6)$$

$$-5 \leq x_i \leq 5, (24\text{bit}), VTR = 10^{-6}$$

この関数は Randomized Sphere model (Rs) であり、Sphere model の解の位置を $x_i = r_i$ とせずらし、対称性を破ってある. すなわち、 $x_1 \neq x_2 \neq \dots, x_N$ である. また、 r_i は、 $[-5, 5]$ の一様乱数である. Sp と同様、単峰性、変数分離可能である.

$$F_{\text{Rd}} = \sum_{i=1}^N \left[\sum_{j=1}^i (x_j - r_j) \right]^2 \quad (2.7)$$

$$-65.536 \leq x_i \leq 65.536, (27\text{bit}), VTR = 10^{-4}$$

この関数は, Schwefel's Double Sum の解の位置を $x_i = r_i$ としてずらしたものであり, Randomized Double Sum(Rd) と呼ぶ. これは, 単峰性, 変数分離不可能, 解は非対称である.

$$F_{\text{Ra}} = 10N + \sum_{i=1}^N [x_i^2 - 10 \cos(2\pi x_i)] \quad (2.8)$$

$$-5.12 \leq x_i \leq 5.12, (24\text{bit}), VTR = 10^{-2}$$

この関数は Rastrigin's function(Ra) と呼ばれる多峰性関数であり, 解は対称的で, $x_i = 0$ で, 最小値 0 をとる. また, この関数は変数分離可能である. 極小値の数は, この定義域の範囲内で, 1次元当たり 11 個ある.

$$F_{\text{Gr}} = \frac{1}{d} \sum_{i=1}^N (x_i - 100)^2 - \prod_{i=1}^N \cos\left(\frac{x_i - 100}{\sqrt{i}}\right) + 1 \quad (2.9)$$

$$d = 4000, -600 \leq x_i \leq 600, (31\text{bit}), VTR = 10^{-4}$$

この関数は Griewank's function(Gr) と呼ばれる多峰性関数であり, 解は対称的で, $x_i = 100$ で最小値 0 をとる. この関数は変数分離不可能である. 極小値の数は, 1次元目ではおよそ 200 個もあり, 最小値のすぐ隣の極小値の値は, $x_1 = 100 \pm 2\pi$ を代入して, およそ 0.01 である.

$$F_{\text{Mi}} = - \sum_{i=1}^N \sin(x_i) \sin^{2m}\left(\frac{ix_i^2}{\pi}\right) \quad (2.10)$$

$$m = 10, 0 \leq x_i \leq \pi, (22\text{bit}),$$

$$VTR = -4.687(N = 5), -9.66(N = 10) \quad (2.11)$$

この関数は Michalewicz' Function(Mi) と呼ばれる多峰性関数である. 解は非対称であり, 最小値は 5次元と 10次元では異なるが, 解析的には解くことができない. この関数は変数分離可能である. 極小値の数は, 1次元目では 1 個, i 次元目では i 個存在する. よって N 次元空間では $N!$ 個の極小値が存在することになる.

$$F_{\text{Sh}} = - \sum_{i=1}^m \frac{1}{\sum_{j=1}^N (x_j - a_{ij})^2 + c_i} \quad (2.12)$$

$$m = 30, 0 \leq x_i \leq 10, (24\text{bit}), VTR = -9 \quad (2.13)$$

この関数は Shekel's foxholes(Sh) と呼ばれる多峰性関数である。解は非対称である。 a_{ij} , c_i はあらかじめ決められた乱数であり, $x_j = a_{ij}$ に極値を持つ Cauchy 分布 (または, Lorentz 分布) を 30 個重ね合わせた関数であるので, 極小値の数は 30 個である。また, この関数は変数分離不可能である。

$$F_{La} = - \sum_{i=1}^m c_i \exp \left[-\frac{1}{\pi} \sum_{j=1}^N (x_j - a_{ij})^2 \right] \cos \left[\pi \sum_{j=1}^N (x_j - a_{ij})^2 \right] \quad (2.14)$$

$m = 5, 0 \leq x_i \leq 10, (24bit), VTR = -1.4$

この関数は Langerman's Function(La) と呼ばれる多峰性関数である。解は非対称である。 a_{ij} , および c_i は上記の Sh での値と同じものを用いる (ただし, c_2 のみ異なる)。この関数は, 多次元の Gauss 分布を包絡線とする振動関数を 5 つ重ね合わせた関数であるが, 振動項があるために極小値は非常に多い。また, この関数は変数分離不可能である。

また, Sp, Gr, Mi, Sh, および La の 5 関数は第 1 回 ICEO[16] で用いられた関数である。Gr, Mi, Sh, La の 4 関数の 2 次元に投影した概形を図 2.6 から 2.9 に示す。ただし, Gr については最小値付近を拡大してある。

遺伝子型は各変数 x_i の定義域に応じて, 式中に示した bit 数に Gray coding したものを用いる。これは, 変数 x_i の量子化精度を 10^{-6} としたものに相当する。

また, 比較のために, SSGA を用いて同様の計算を行う。SSGA は前章で述べた通り, 一定の集団数のもとで, 交叉, 突然変異, 選択を繰り返すアルゴリズムである。

1 回の試行における終了条件は, PfGA と SSGA で対等に比較するため, 10,000 回の評価 (個体の適応度を評価する回数) とする。SSGA の遺伝的パラメータは, $(T, M, p_m) = (10000, 100, 0.02)$ である。また, 関数の次元はここでは 5 とする。

2.5 実験 2 に対する結果

前節で示した 9 関数に対し, SSGA, および PfGA の, 1 回の試行における最良個体の関数値を示す。まず図 2.10 は 5 次元 Sp に対する探索の状況である。実線が SSGA, 破線が PfGA による結果であり, 横軸は評価回数, 縦軸は関数値である。探索状況は順調であり, 収束の速さでは PfGA の方が優っている。

図 2.11 は 5 次元 Ds に対する探索状況である。この場合も探索は順調であり, PfGA の方が速いことが分かる。

図 2.12, 2.13 は 5 次元 Rs, Rd に対する探索状況である。この場合も探索は順調であり, PfGA の方が速いことが分かる。

図 2.14, 2.15 は 5 次元 Ra, Gr に対する探索状況である。これらの関数は多峰性関数であるため, 一般に局所解に陥りやすいとされている。図中で, 関数値の変化しないある一定のプラトー (plateau) が存在することが確認できるが, これらが局所解に陥った状況を表している。しかしながら最終的には最適解の方の谷へ移行し, 探索は成功している。収束速度に関しては, 両アルゴリズムはほぼ互角である。

図 2.16, 2.17, および 2.18 はそれぞれ, 5 次元 Mi, Sh, および La に対する探索状況である。これらの関数も多峰性関数であり, 極値に陥りやすいのであるが, 両アルゴリズム共に探索

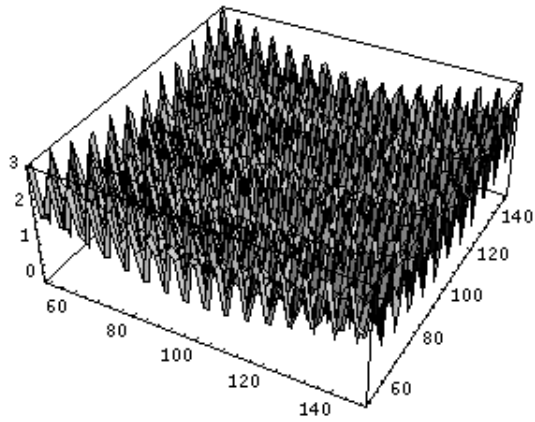


図 2.6: Griewank's Function

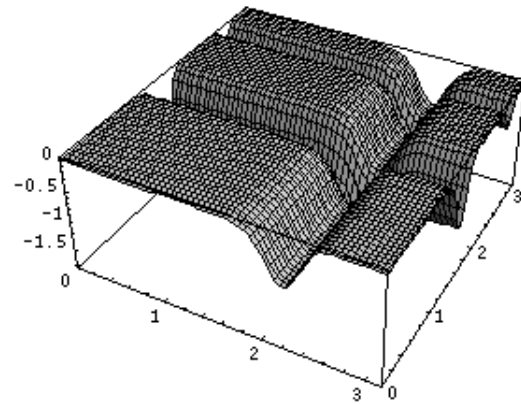


図 2.7: Michalewicz' Function

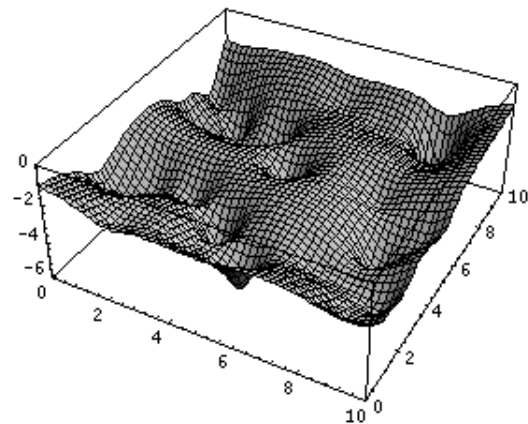


図 2.8: Shekel's foxholes

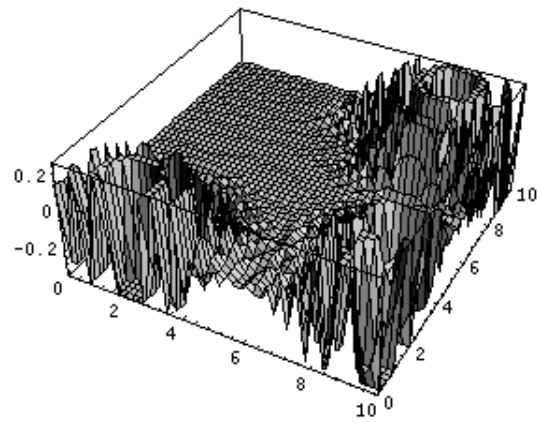


図 2.9: Langerman's Function

は成功している. 収束速度に関しては, Mi に対してはほぼ互角であり, Sh に対しては PfGA の方が優れている. La に対してはわずかながら SSGA の方が速いようである.

以上の結果をまとめると, ほぼ全ての関数で, PfGA の収束の速さが確認できた. また, 単峰性関数のように比較的簡単な関数に対しては, SSGA, PfGA とともに指数関数的な収束状況であるのに対し, 多峰性関数についてはさまざまな収束の仕方をしていることが分かる. これらの結果は全て 1 回のみを試行であり, 初期値も異なっている点を考慮しておかなければならない. SSGA では初期個体数が 100 であり, PfGA では 1 であるため, 同等の初期状態を準備することは不可能である.

SSGA では, 設定した 1 組のパラメータを全ての関数に対して用いたが, 設定次第では勿論, PfGA よりも優れた性能となる可能性がある. しかしながら, パラメータの設定をする場合はそれだけで非常に多くの労力と時間を費さなければならない点を考えると, PfGA は実に汎用的であると言える.

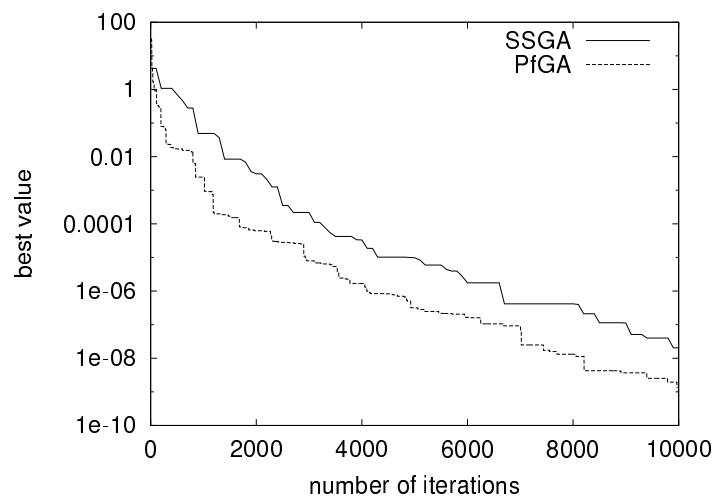


図 2.10: 5次元 S_p に対する, SSGA, および PfGA による最良個体の関数値

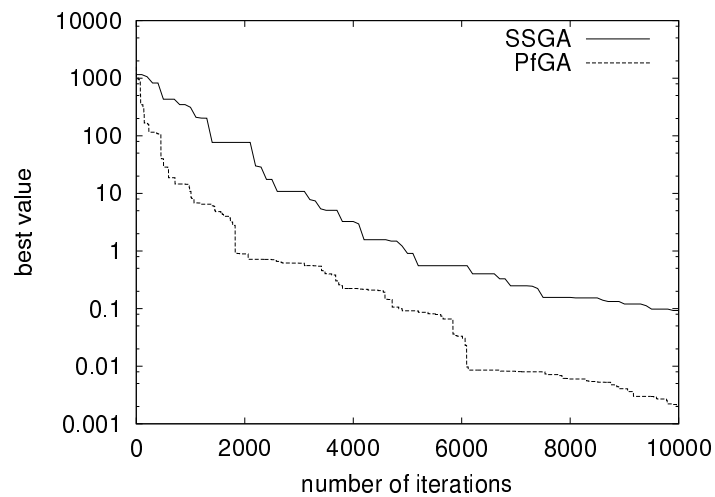
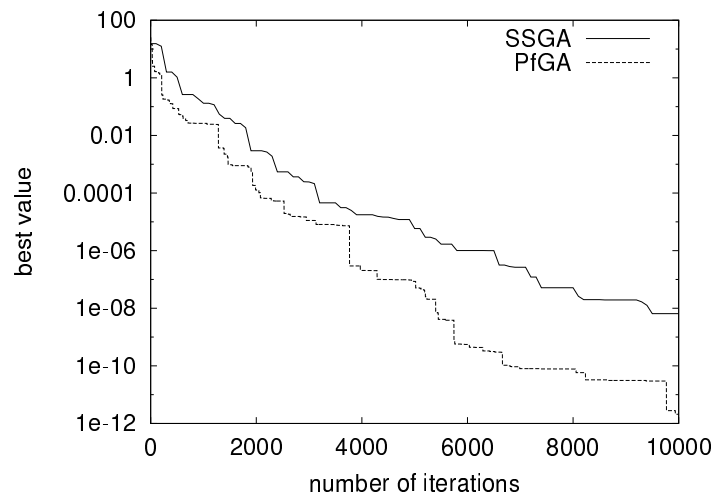
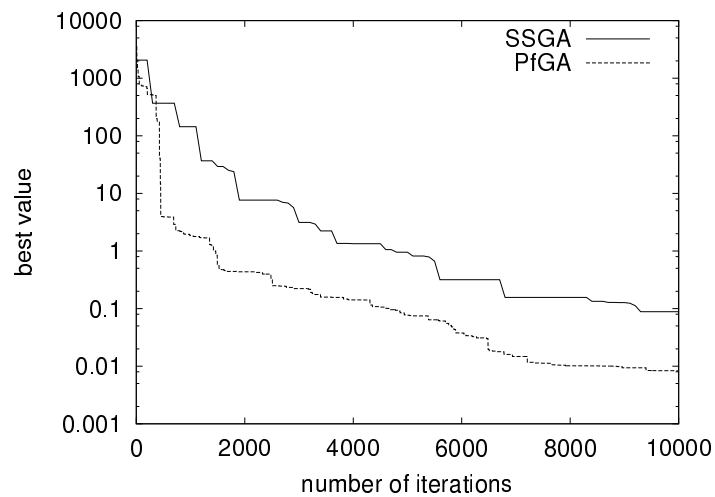


図 2.11: 5次元 D_s に対する, SSGA, および PfGA による最良個体の関数値

図 2.12: 5次元 R_s に対する, SSGA, および PfGA による最良個体の関数値図 2.13: 5次元 R_d に対する, SSGA, および PfGA による最良個体の関数値

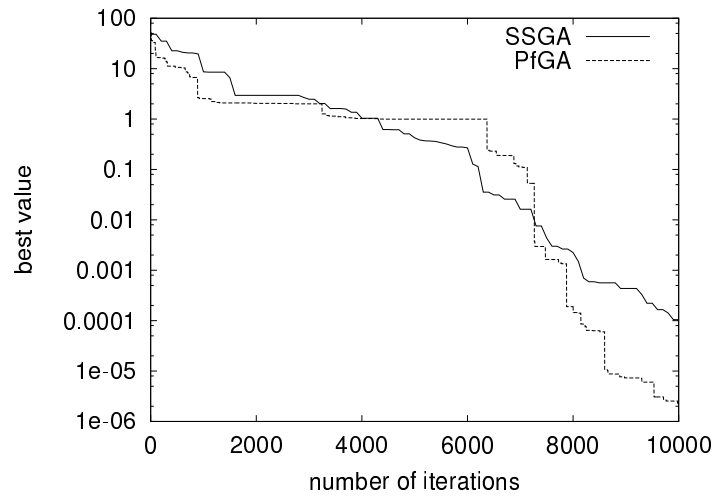


図 2.14: 5次元 Ra に対する, SSGA, および PfGA による最良個体の関数値

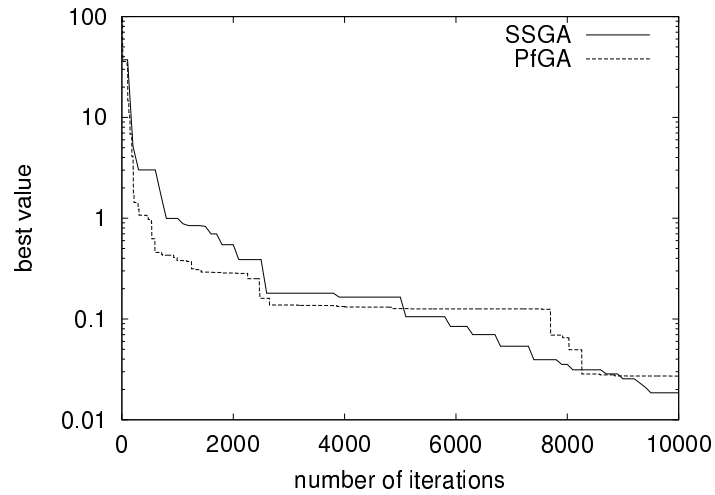


図 2.15: 5次元 Gr に対する, SSGA, および PfGA による最良個体の関数値

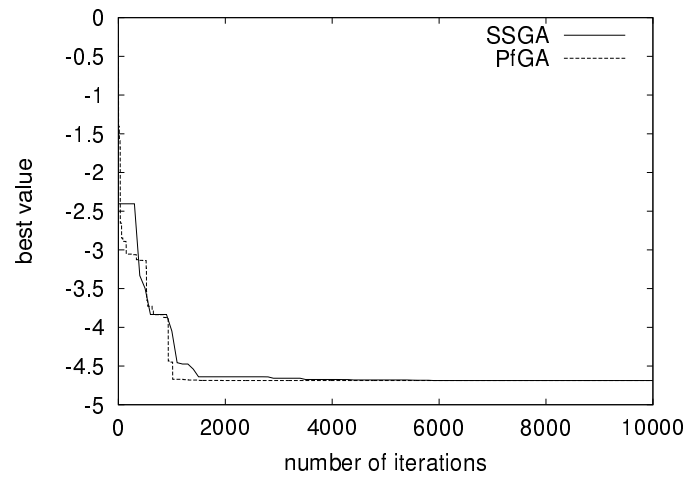


図 2.16: 5次元 Mi に対する, SSGA, および PfGA による最良個体の関数値

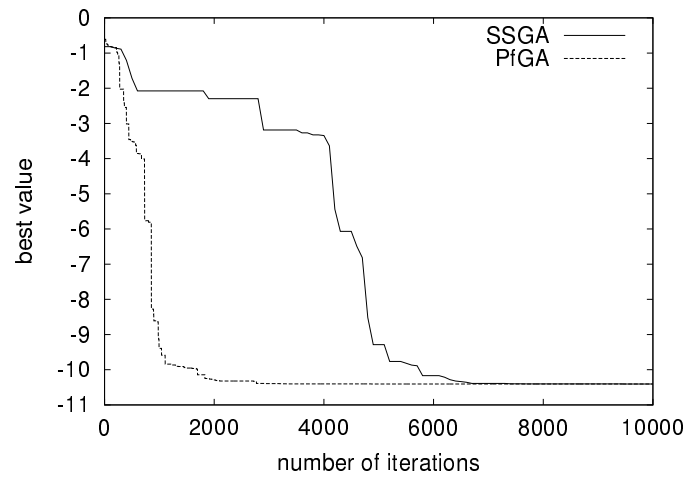


図 2.17: 5次元 Sh に対する, SSGA, および PfGA による最良個体の関数値

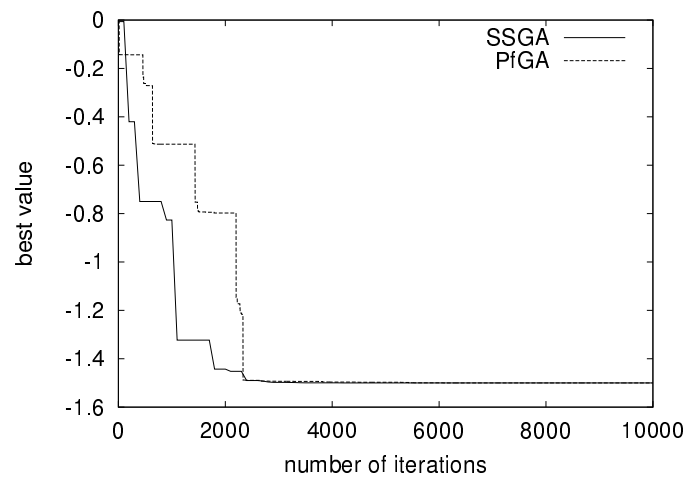


図 2.18: 5次元 La に対する, SSGA, および PfGA による最良個体の関数値

2.6 実験 3: 平均評価回数の評価と ICEO の他のアルゴリズムとの比較

近年, 数多くの進化的計算アルゴリズムが提案されてきているが, これらの優劣を同一基準で比較することが急務となってきた. このような背景から, 進化的最適化に関して, 1996年に第1回国際コンテスト (International Contest on Evolutionary Optimisation: ICEO) が開かれた [16]. 最小値探索を行う目的関数は前述の Sp, Gr, Sh, Mi, および La である.

GA では, 従来から, オンラインパフォーマンスや, オフラインパフォーマンスなどの, 1回の試行における評価基準を用いてきたが, ICEO では, アルゴリズムの性能を評価する指標として, BV, ENES, および RT という量を統計的に求めている. BV(Best Value) は, 20回試行中で最も良い個体の関数値である. また, ENES (Expected Number of Evaluations per Success) は, 1成功当たりの平均評価回数である. すなわち, 1回の試行において VTR に到達すれば成功とみなし, その VTR に到達するまでの評価回数を, 成功試行で平均したものである. RT(Relative Time) は, 適応度 (関数値) の評価を除いたアルゴリズム自体を実行するための相対時間であり, 次式で定義されている.

$$RT = \frac{CT - ET}{ET} \quad (2.15)$$

ただし, ET は, 評価関数を 10,000 回評価するのに要する CPU 時間, CT は, アルゴリズム全体の CPU 時間である. RT が小さければ, 評価以外の計算 (例えば関数値のスケーリング等) が少なく, コンパクトなアルゴリズムであることを意味している.

コンテストに参加したアルゴリズムの内上位5つと, SSGA, PfGA の結果を, 5次元版について表 2.1 に併記する. ただし, 表中の左欄の5関数の後ろの数字は次元を表す. この表より, PfGA は全体的に SSGA に比べ, 同等かそれ以上の収束性能を持っていると見ることができる. その他のアルゴリズムと比較した場合でも, ENES は2位, または3位という成績となった. また, RT についても PfGA は1位, 2位または3位となり, 計算量を抑えているアルゴリズムであることがわかる.

2.7 実験 4: 成功率の導入と PfGA, SSGA の評価

ICEO では, アルゴリズムの性能を評価する指標として BV, ENES, RT の3つを用いた. ここではさらに統計的に信頼できる評価基準を考える. 独立した試行を数多くこなすことによって, BV(Best Value) の値はさらに小さくなるであろう. しかしながら, 一旦 VTR(Value To Reach) が与えられた問題に対しては, そのボーダーラインを越えたかどうかはアルゴリズムの優劣を判定する材料となるはずである. そこで, VTR を越えた試行を成功したものとみなし, 全試行回数に対する比率を成功率 (Success Rate: SR) と呼ぶことにし, これを BV の代わりとして用いることにする. 成功率を用いる利点としては, 統計的評価が容易となる他に, 問題が GA にとって容易であるか困難であるかという目安にもなる点が挙げられる. さらに, 最適化アルゴリズムの本来の目的である収束の速さの他に, 確率的探索アルゴリズムの課題とも言える, 与えられた時間内にどのような初期値を用いても必ず問題を解けるという安定性 (ロバスト性) の評価にもつながると考えられる.

表 2.1: 第1回ICEOとSSGA, PfGAの結果

ENES BV RT	Bi-Pa [17]	Li [18]	Sto-Pri [19]	Van Ke [20]	Se-Be [21]	SSGA	PfGA
Sp5	20	243	736	1,452	326	6,204	4,067
	3.88e-15	0.0	-	1.5e-30	-	5.6e-10	0.0
	2	12.7	4.67	75	0.35	18.27	0.91
Gr5	41	21,141	5,765	22,039	35,637	10,558	6,785
	7.99e-6	1.69e-5	-	3.2e-12	-	8.9e-14	4.66e-7
	2	3.1	1.79	2	0.35	14.59	0.90
Sh5	74	6,318	76,210	51,845	13,836	7,734	1,619
	-10.327	-10.403	-	-9.83	-	-10.404	-10.40
	2	0.25	0.80	0.96	0.35	19.81	0.33
Mi5	120	6,804	1,877	10,661	9,925	4,190	5,131
	-4.6876	-4.687	-	-4.687	-	-4.68766	-4.688
	2	1.28	1.11	3.9	0.35	18.48	0.90
La5	176	4,131	5,308	11,343	74,720	3,098	5,274
	-1.499	-1.499	-	-1.49	-	-1.5	-1.499
	2	1.62	1.35	1.8	0.35	15.92	0.43

そこで、次の実験は、ICEO5関数の5次元と10次元に対して、それぞれ独立に300回の試行を行い、成功率SR、平均評価回数ENESの統計量を求める。また、アルゴリズムは同様にSSGAとPfGAを使用する。SSGAに用いるパラメータは個体数を100、突然変異率を0.02と0.05の2種類とする。また、1回の試行における終了条件である評価回数は、5次元で10,000回、10次元で100,000回とする。

結果は表2.2のようになる。SSGAのパラメータである突然変異率が異なると結果は大幅に異なることが分かる。Spでは、SRが100%と0%であり、他の関数に対しても、突然変異率が0.05の時は探索が失敗に終わっているケースが多い。

このように、SSGAが職人芸のようにパラメータを調節しなければならないのに対し、PfGAではそのパラメータを設定せずとも、右欄のような優秀な結果を出している。PfGAでは、失敗したケースがなく、より安定したアルゴリズムであると言える。また、注目すべき点として、PfGAの収束の速さが挙げられる。ほとんどの関数に対して、ENESがSSGAと同等かそれ以下となっており、優れた収束性能を持っていることが分かる。

2.8 考察: 選択における場合分け

PfGAの内部状態はどのようになっているのであろうか。PfGAにおける選択は変則的であり、親子4個体の適応度の大小を4つのCaseに場合分けしている。そこで、それぞれの

表 2.2: SSGA と PfGA の SR と ENES

Func	SSGA (0.02)		SSGA (0.05)		PfGA	
	SR(%)	ENES	SR(%)	ENES	SR(%)	ENES
Sp5	100	6,299	0	-	100	4,002
Gr5	0.67	9,611	0	-	0.33	9,670
Sh5	0	-	0.67	3,836	7.67	1,758
Mi5	40.33	4,875	26	7,957	29.67	3,170
La5	84.33	3,851	94	4,663	38	4,159
Sp10	100	25,961	0	-	100	12,512
Gr10	20	43,675	0	-	0.67	10,446
Sh10	0	-	0	-	1.33	4,113
Mi10	2.67	41,739	0	-	2.67	45,731
La10	25.67	39,150	40	62,584	1	77,118

表 2.3: PfGA の選択における各ケースの割合

Func	Case 1	Case 2	Case 3	Case 4
Sp5	0.01	4.38	91.31	4.30
Gr5	0.02	7.38	87.83	4.77
Sh5	0.02	12.70	83.09	4.19
Mi5	0.02	4.62	91.37	3.99
La5	0.02	29.20	68.28	2.50

Case がどの程度の頻度で出現しているのかを、ICEO5 関数の 5 次元に対して、それぞれ 300 回試行してその割合を出したものが表 2.3 である。

この表において、Case 1 の割合は非常に小さく、2 つの子個体が共に親個体よりも良いというケースは稀であることが分かる。また、探索が成功するケースは Case 1 と Case 4 の和であるが、この割合が目的関数によって異なり、適応的に探索を行っていることの一つの現れとなっている。

アルゴリズムの説明の箇所では省略したが、Case 3、および Case 4 はさらに細かく分類することができ、改良すべき点はまだ残されていると考えられる。

2.9 まとめ

本章では、GA における初期集団数、交叉率、突然変異率などの遺伝的パラメータの設定を行う必要がない PfGA について述べてきた。PfGA は、family 内の個体の適応度の大小に応じて 1~3 個体を選択し、大局的な探索と局所的な探索のバランスを動的に取りながら最適解を見つけ出していくアルゴリズムである。

アルゴリズムの性能を評価する指標として、従来は、オンラインパフォーマンスやオフラインパフォーマンス等が用いられてきたが、ICEO では、BV, ENES, RT の3つでアルゴリズムの比較を行った。また、より統計的に信頼性のある、成功率 (SR) を定義し、これを新たな尺度として評価を行った。PfGA は、9 関数の範囲内では、SSGA と比較しても収束が速く、かつ安定したアルゴリズムであることが分かった。これは、SSGA が集団ベースのアルゴリズムであるのに対し、PfGA は少数個体を用い、無駄な計算を極力省いたために速くなったのであり、また、少数個体 GA の欠点である初期収束の問題も、適応的な探索によって解決されているのである。

ICEO5 関数の範囲内においても、PfGA は他の手法と比べて収束性能が見劣りすることなく、良い結果を得た。PfGA は、さまざまな意味での拡張性を持ったアルゴリズムであると言える。

また PfGA では、突然変異の仕方が2つの子個体で異なっている。これは、古澤らの、DNA の2重鎖 (double strand) に生じる突然変異による不均衡進化論 (disparity evolution)[22] に合致している考え方である。不均衡進化論によると、生物において、DNA の2重鎖 (2倍体とも言う) がほどける向きと、コピーされる向きの等しいリーディング鎖 (leading strand) と、これらの向きが逆であるラギング鎖 (lagging strand) とで、突然変異率が異なる。すなわち、リーディング鎖にはほとんど突然変異は生じないが、ラギング鎖には突然変異が生じやすい。前者は集団の安定性 (stability) に寄与し、後者は柔軟性 (flexibility) に寄与するとしている。

このような、生物が本来持っている進化や遺伝の絶妙なバランスは、PfGA のように積極的に採り入れるべきである。しかしながら、PfGA には問題点もある。選択、淘汰に関して、なぜこのようなルールで優れた探索能力を持つのかという点である。すなわち、1~3 個体を選ぶ際に、親子4 個体の適応度の大小から Case 1~4 の場合分けをしているが、これは試行錯誤の上で採用したものであり、理論的に最適な方法であると証明されたわけではない。この証明は非常に困難であるが、今後の課題としたい。

第3章 パラメータフリー遺伝的アルゴリズムの並列分散処理の提案と性能評価

3.1 概要

本章では、探索速度の向上を目的とし、PfGA の並列分散処理 [23] について述べる。探索を速めるために局所探索を重視すると局所解に陥ってしまい、なかなかそこから抜け出せなくなる可能性がある。そこで、GA の大域的探索能力をそのまま引き継ぎ、なおかつ収束性能に優れたアルゴリズムを提案する。

GA の並列分散処理の研究は広範囲に行われている [24] [25][26][27][28] [29]。GA を並列分散処理するメリットは、処理の高速化の他に、探索空間を分割したり融合することで初期収束を防止し、最適解への到達確率を向上できる点である。並列化の方法としては、大別して処理単位とプロセッサの割り当ての仕方、すなわち粒度 (granularity) により、粗粒度 (coarse-grained)[24] [27][28] から密粒度 (fine-grained)[25] [26][29] までのレベルがある。

PfGA を用いる理由は、粗粒度並列分散処理の概念と、少数個体で構成される PfGA の局所集団の並列化という発想との整合性からである。すなわち、複数の局所集団を並列化する際、パラメータは1つ増え、より複雑になることが予想されるため、できるだけ汎用的でパラメータ設定の必要のない GA を用いる方が容易であるからである。

本章の主な流れは、PfGA の粗粒度の並列分散アーキテクチャについて検討し、4 種類の実装方法を考案する。また実験は、並列化された計算機上で行い、平均評価回数、成功率などの統計量を求め、これによって、PfGA の持つ優れた探索能力をさらに強化できる可能性を示す。

3.2 遺伝的アルゴリズムの並列化

一般に、処理を並列化する場合の目的は処理速度の高速化であるが、特に探索問題の場合には、次に示すように特別な状況が起こり得る。図 3.1 は、10 個の探索点を逐次処理により探索する場合と、並列処理により探索する場合の模式図である。7 番目の点が最適解であるとし、探索は左から順に行うものとする。逐次処理では7回目に解を得るが、並列度を2として並列処理を行うと2回目に解を得ることができる。すなわち合計して4回の計算で解へ到達したことになるのである。

与えられたタスクを並列化する場合は、並列度を N とすると、通信速度のような余分な時間は無視するとしても、処理時間は $1/N$ となるのに対し、ここに示した例は特殊ではあ

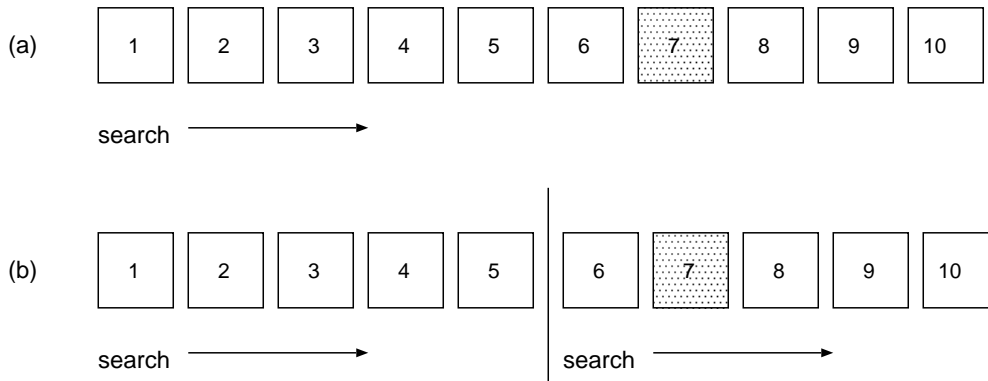


図 3.1: 逐次探索 (a) と並列探索 (b)

るが、探索問題に対しては、工夫次第で処理をより速くすることができるのである。

GAによって探索を行う場合は、移民による個体間の相互作用によって、より効率良く探索出来る可能性がある。粗粒度の並列GAの場合は、局所集団を処理単位にし適度な頻度で個体を局所集団間で移民 (migration) させることが多く、密粒度の場合は、ある個体の近傍を処理単位として近傍間に重なりを持たせる場合が多い。前者は、島モデル (island model) とも呼ばれ、一つの局所集団 (島) が一つの生物種の最小交配単位 (deme) を構成している。ここでは実装の容易な前者のモデルを用いる。

さらに並列処理には同期、非同期の区別があり、非同期の場合、各CPUの負荷状況がダイナミックに変化し、各試行における再現性は保証されない。このような場合、問題とするアーキテクチャの違いによる効果とは別の本質的でない部分 (例えば、移民のタイミングのずれなど) が関与し、並列の効果を正当に評価することができなくなると考えられる。よって実験の再現性の観点から同期的に処理を行うものとし、 M 個のCPUに N 個の局所集団を割り当てて同期的に進化させ、異なる局所集団間で個体の移民を行わせる [23]。

次に実装方法として、2種類のアーキテクチャと、2種類の移民方法について述べる。

3.2.1 アーキテクチャ：一様分散型とマスタースレーブ型

並列分散アーキテクチャとしては、一様分散型とマスタースレーブ型の2種類を考える。一様分散型アーキテクチャとマスタースレーブ型アーキテクチャの違いは、前者が全ての局所集団が同じ役割を持つのに対し、後者は局所集団全体を監視するマスターの局所集団を一つ有する。

図3.2にPfGAの一様分散型、およびマスタースレーブ型アーキテクチャを示す。 S は全探索空間、 S' は局所集団を表す。それぞれの局所集団 S'_i では、前述のPfGAの交叉、突然変異、選択を行う family S''_i が存在し、選択された個体は再び局所集団 S'_i に戻される。

マスタースレーブ型では、 S'_0 をマスターの局所集団、 S'_i ($i = 1, \dots, N$)をスレーブの局所集団とする。マスター S'_0 は、常に (あるいは一定間隔で) 全てのスレーブ中の最良個体を監視し、あるタイミングで移民してきた個体を処理したり、その他のスレーブへ個体を分配し

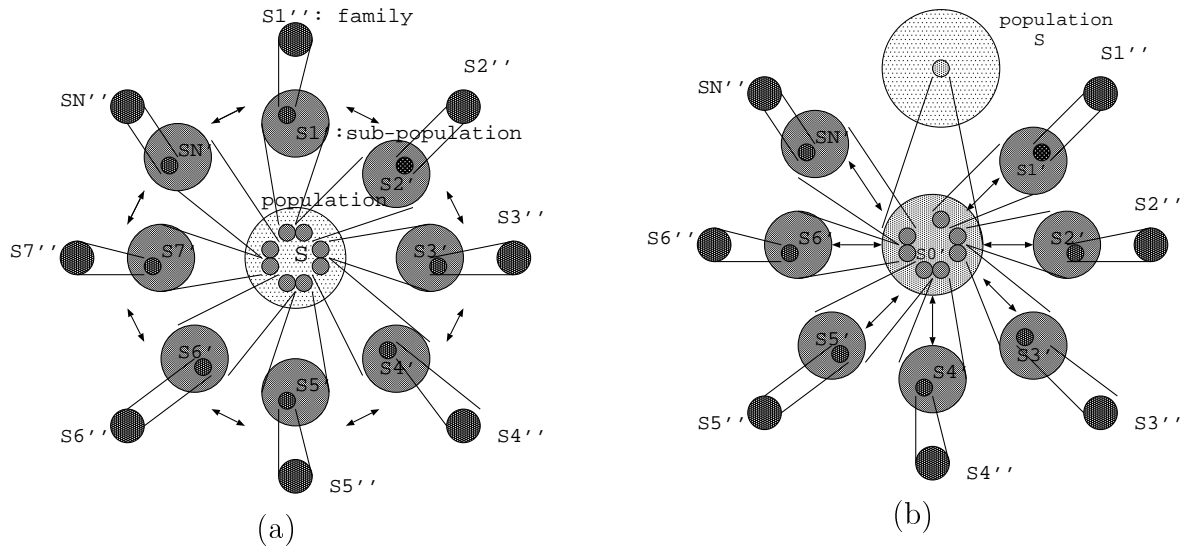


図 3.2: 一様分散型アーキテクチャ(a)とマスタースレーブ型アーキテクチャ(b). 矢印は個体の移動を表す

たりする.

また, 図中の矢印は, 局所集団間で「移民」が起ることを表す.

3.2.2 移民方法: 直接移民型と階層移民型

移民のさせ方としては種々の方法が考えられるが, 1つ目は, ある局所集団で良い個体が生じた場合にのみ, 他の局所集団にコピーして分配するという方法である. これを直接移民型と呼ぶことにする. ただこの方法では, 移民後に他の局所集団でも同じ個体を保持することになり, システムの多様性がなくなる可能性がある. そこで2つ目として, 良い個体を複数の局所集団から集め, これらの中から任意の2個体を新しい親個体として選択する. そして, 交叉と突然変異によって2個の子個体を生成し, PfGAの選択則に従って1~3個体を任意に選択した局所集団に分配する方法が考えられる. これは局所集団レベルからみるとメタレベルのPfGAを用いた移民方法であるので, 階層移民型と呼ぶことにする.

3.3 4つの実装方法によるアルゴリズムの構築

以上のように, 並列アーキテクチャとしては一様分散型とマスタースレーブ型の2種類, 移民方法としては直接移民型と階層移民型の2種類を組合せ, 計4種類の方法で並列処理を試み, それぞれの移民の効果を調べる. この4つの方法をまとめると次の通りである.

- 一様分散/直接移民型 (UD1)

図 3.3(a) に示すように, ある局所集団 S'_i ($i = 1, \dots, N$) において, Case 1 または 4 で生じた最良子個体 C_1 をその他の任意の局所集団 S'_j ($j \neq i$) にコピーして分配する. 分配された局所集団 S'_j では, 集団中で最悪個体を淘汰して個体数を元に戻す.

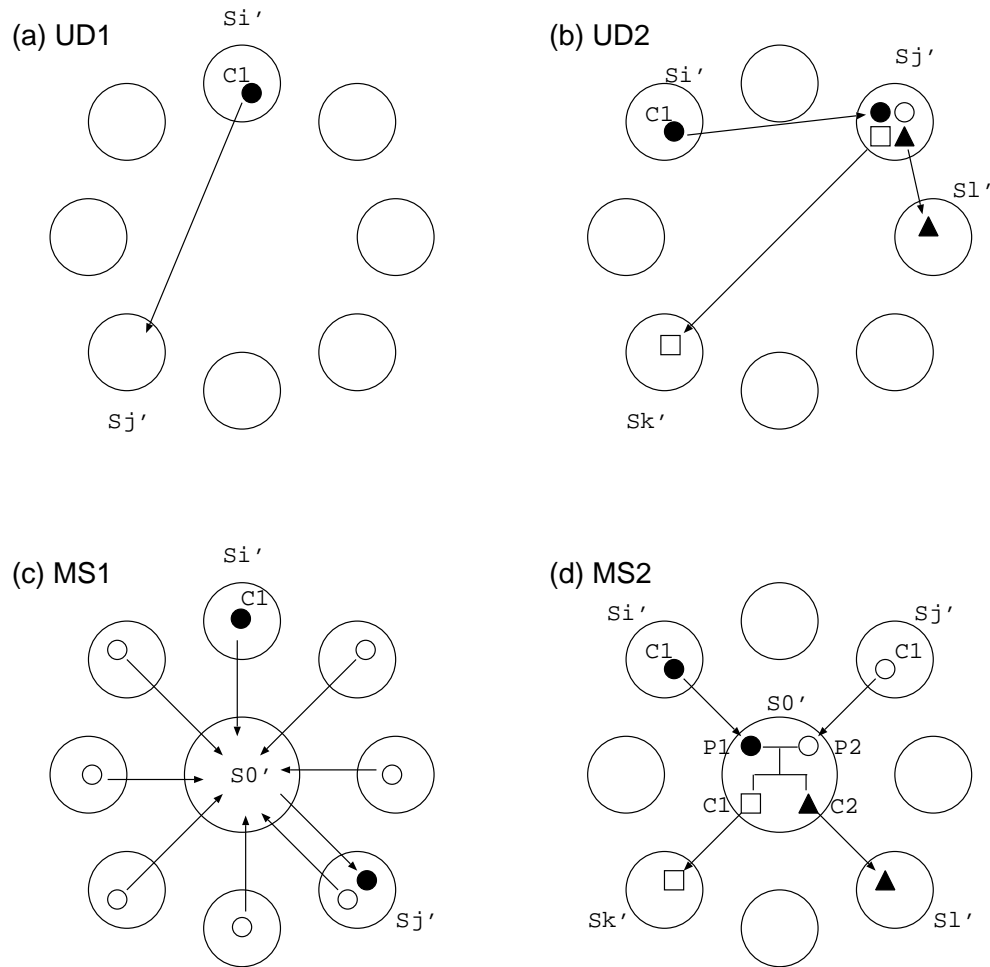


図 3.3: 移民選択法: UD1(a), UD2(b), MS1(c), および MS2(d)

- 一様分散/階層移民型 (UD2)

図 3.3(b) に示すように, ある局所集団 S_i' ($i = 1, \dots, N$) において, Case 1 または 4 で生じた最良子個体 C_1 をその他の任意の局所集団 S_j' ($j \neq i$) にコピーして分配する. 一方, 局所集団 S_j' では, 分配された個体が 2 個以上になった時点でそのうち 2 個体を親 P_1, P_2 とし, 交叉, 突然変異により 2 個の子個体 C_1, C_2 を生成する. 親子 4 個体を PfGA の選択則を用いて選択し, 1~3 個体を任意に選んだ局所集団に分配する. この時, これらの個体は S_j' に元からある個体と直接的には混ざらないが, 別のタイミングで他の局所集団 (例えば, S_k') から分配された個体と S_j' に元からあった個体とは間接的に混ぜ合わされ進化していく. 分配された局所集団では最悪個体を淘汰する. この方法では, 新たに 2 個体を発生したわけであるから, 移民によって評価回数は 1 局所集団当たり 2 増加する.

- マスタースレーブ/直接移民型 (MS1)

図 3.3(c) に示すように, 局所集団 S_0' をマスター, S_i' ($i = 1, \dots, N$) をスレーブとし, スレーブにおいて Case 1 または 4 で生じた最良子個体 C_1 をマスターに分配する. マ

スターは、スレーブから分配される個体のうち最良の個体を選択し、任意に選んだスレーブに分配する。分配されたスレーブでは、移民を含めた集団中から最悪個体を淘汰する。

- マスタースレーブ/階層移民型 (MS2)

図 3.3(d) に示すように、スレーブ S'_i ($i = 1, \dots, N$) において Case 1 または 4 で生じた最良子個体 C_1 をマスター S'_0 に分配する。マスターはスレーブから分配される個体が 2 個以上になった時、そのうち 2 個体を親 P_1, P_2 とし、交叉、突然変異により 2 個の子個体 C_1, C_2 を生成する。親子 4 個体を PfGA の選択則を用いて選択し、1~3 個体を任意に選んだスレーブに分配する。分配されたスレーブでは、集団中で最悪個体を淘汰して個体数を元に戻す。この方法では、新たに 2 個体を発生したわけであるから、移民によって評価回数はマスターの局所集団で 2 増加する。

3.4 実験と結果

実験は、第 2 章で用いた 9 つのテスト関数の 5 次元と 10 次元について、前述の 4 種類のアーキテクチャを用いて、局所集団数 N が 1, 2, 4, 6, 8, 16, 32, 64, 80, 100 と増加した場合についてそれぞれ 300 回の独立した試行を行い、成功率 SR, 平均評価回数 ENES の統計量を評価する。並列 GA における成功率の定義は、 N 個の局所集団のうち、どれか 1 つでも VTR に到達すれば成功したとみなす。

1 回の試行における終了条件は、5 次元で 10,000 回評価、10 次元で 100,000 回評価とし、局所集団当たりの評価回数は $10,000/N$ とした。もちろん、局所集団当たりの評価回数を 10,000 回としても良いと思われるが、同期的処理である以上、同じ計算がシングルプロセッサでの逐次処理で行えることを考慮すると、並列化する場合と並列化しない場合の比較も行わなければならない。従って、全評価回数を一定としたのである。

なお、計算機環境は LAN で結合された DEC Alpha 8 台を PVM Software[30] を用いて並列化した。

実験結果を大まかに述べると、局所集団数 N の増加に伴い、成功率 SR は単調減少する場合と、ある N の値で極大値をとる場合の 2 つのパターンに分かれた。また探索効率を表す ENES はほぼ $1/N$ で減少し、並列処理の効果が現れた。

3.4.1 Sphere model に対する結果

図 3.4, 3.5 は 5 次元 S_p に対する成功率 SR と平均評価回数 ENES であり、図 3.6, 3.7 は 10 次元 S_p に対する成功率と ENES である。横軸は並列度、すなわち局所集団数であり、 $N = 1$ はオリジナルの PfGA での結果である。図より、成功率は単調に減少し、ENES はほぼ N に反比例していることが分かる。成功率が減少するのは、1 局所集団当たりの評価回数の上限が $1/N$ となるためであり、これによって ENES が減少するのである。

また、アーキテクチャ/移民方法に関しては、成功率、ENES 共に UD1 が最も良い結果となり、以下 MS1, MS2, UD2 となっている。UD2, MS1, および MS2 では、 $N = 10$ 付近で成功率が 0% となるため、ENES も $N = 10$ 付近までしかプロットできなかった。同様にして

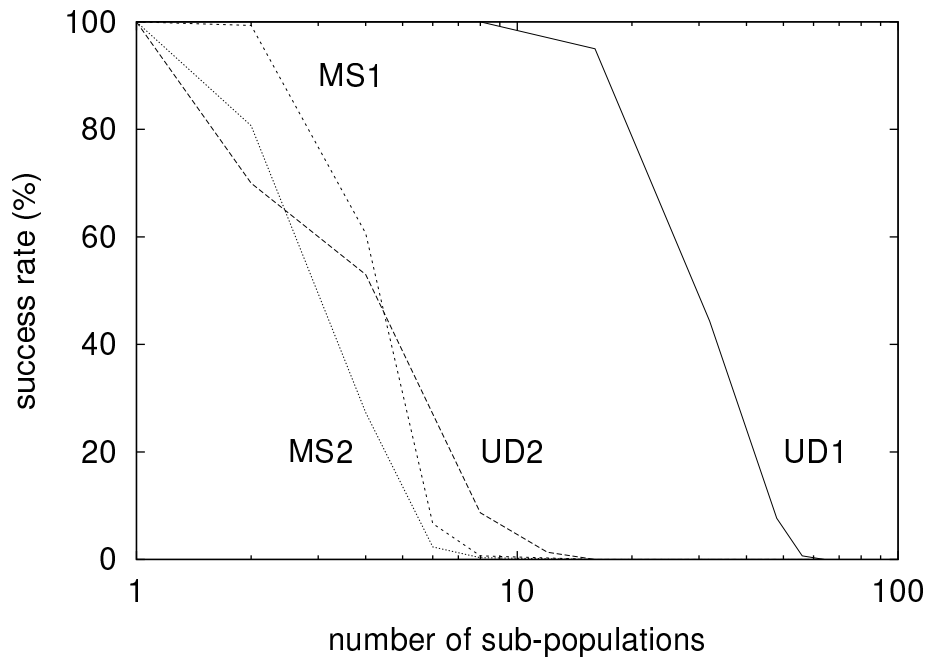


図 3.4: 5次元 S_p に対する成功率

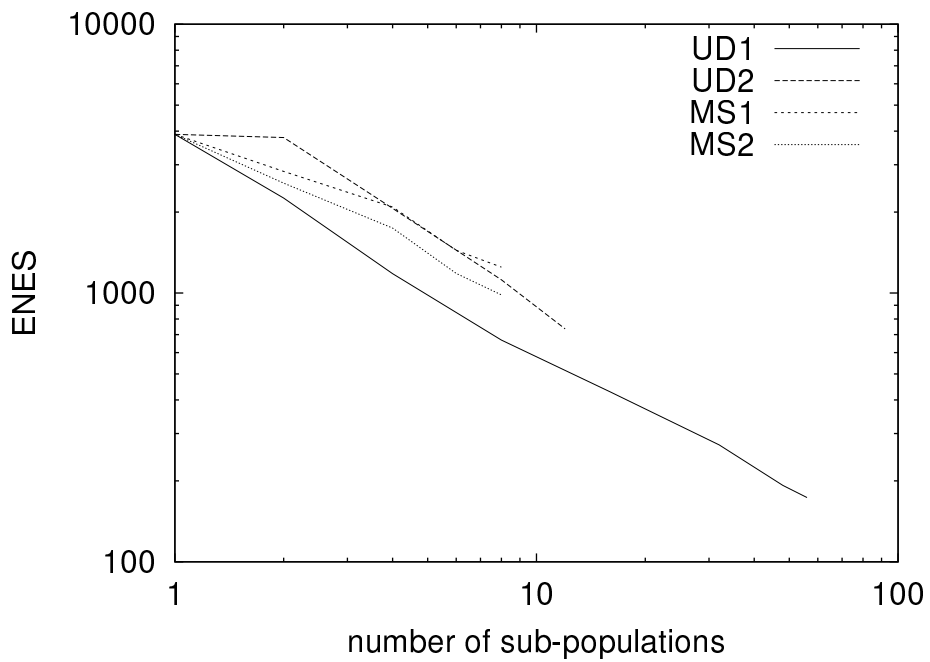
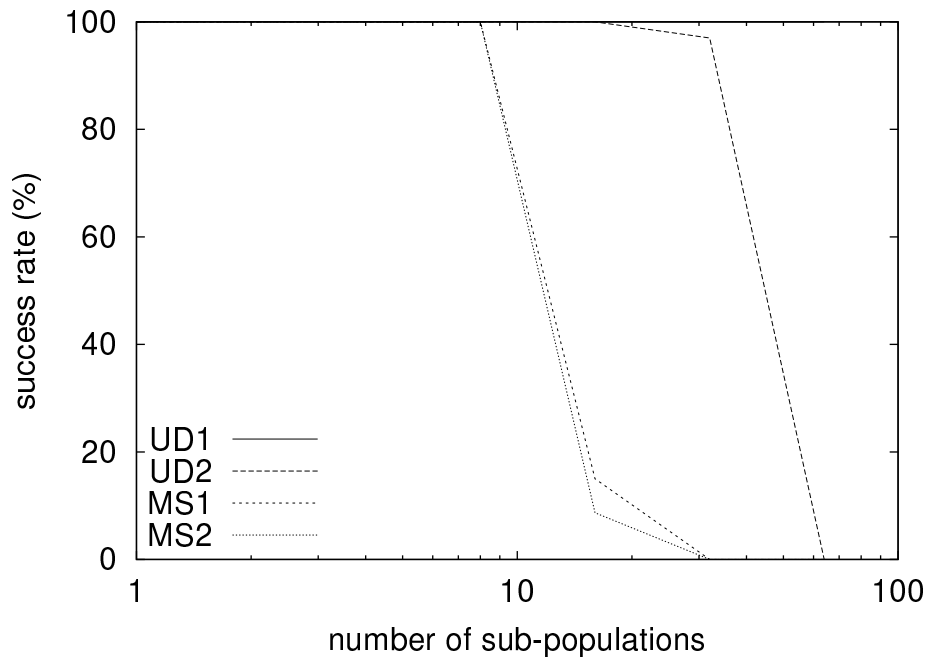
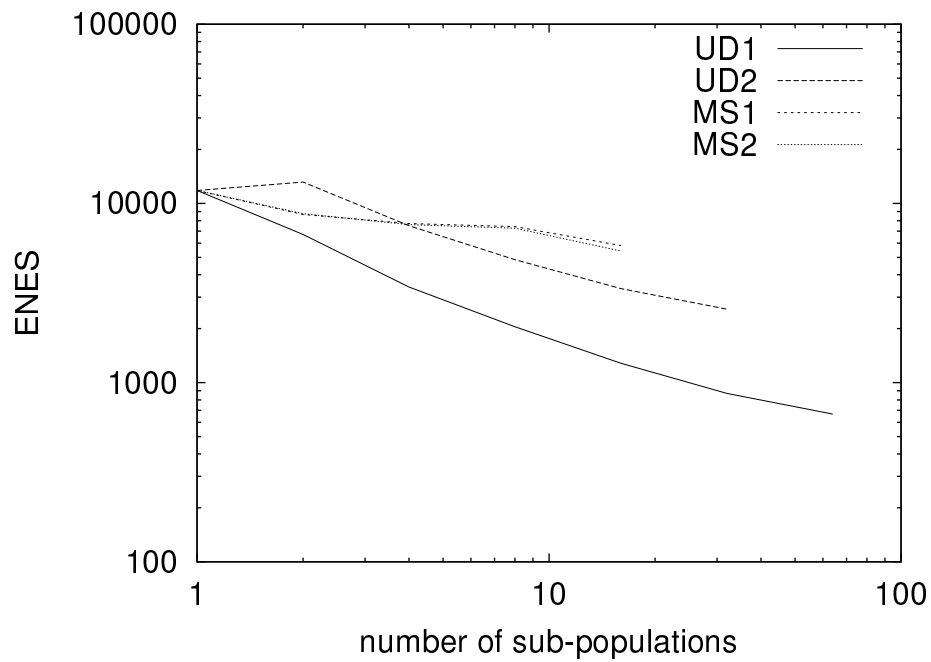


図 3.5: 5次元 S_p に対する ENES

図 3.6: 10次元 S_p に対する成功率図 3.7: 10次元 S_p に対する ENES

10次元の場合もUD1が最も良い結果となった。

3.4.2 Double Sum に対する結果

図3.8, 3.9は, 5次元Dsに対する成功率SRと平均評価回数ENESであり, 図3.10, 3.11は, 10次元Dsに対する成功率SRと平均評価回数である。図より, 成功率は単調に減少し, ENESはほぼ N に反比例している。また, アーキテクチャ/移民方法に関しては, 成功率, ENES共にUD1が最も良い結果となり, 以下, MS1, MS2, UD2の順となった。同様にして10次元の場合もUD1が最も良い結果となった。

3.4.3 Randomized Sphere model に対する結果

図3.12, 3.13は, 5次元Rsに対する成功率とENESであり, 図3.14, 3.15は, 10次元Rsに対する成功率とENESである。図より, 成功率は単調に減少し, ENESはほぼ N に反比例している。また, アーキテクチャ/移民方法に関しては, 成功率, ENES共にUD1が最も良い結果となり, 以下, MS1, MS2, UD2の順となった。同様にして10次元の場合もUD1が最も良い結果となった。

この関数の性質は, S_p の最小値の座標をずらしたものであり, 変数に対する対称性が破れているが, この非対称性が結果に及ぼす影響はそれ程ではないことが, 図3.4-3.7の外形との比較から分かる。

3.4.4 Randomized Double Sum に対する結果

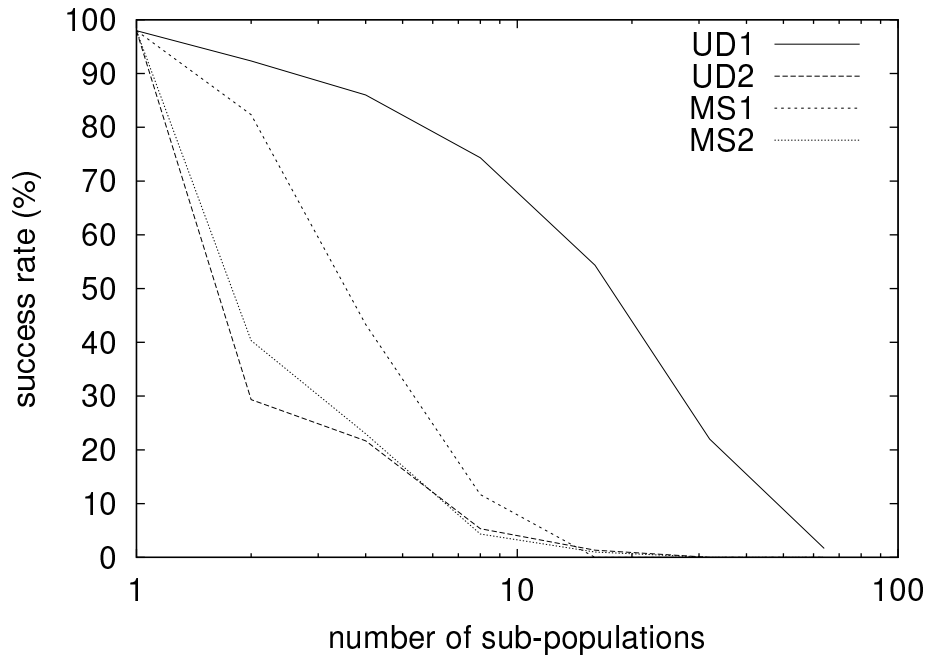
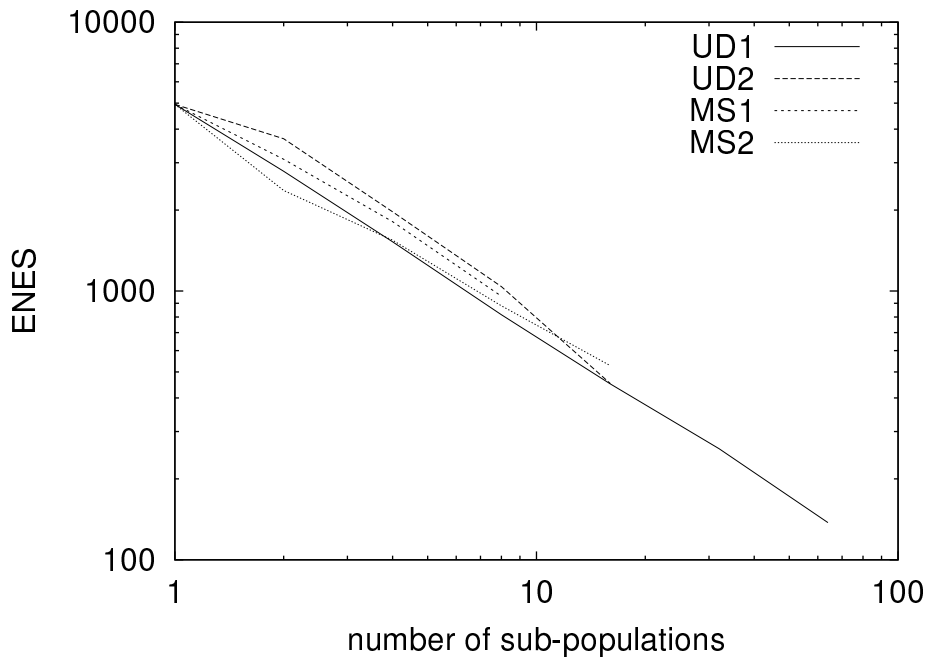
図3.16, 3.17は, 5次元Rdに対する成功率とENESであり, 図3.18, 3.19は, 10次元Rdに対する成功率とENESである。図より, 成功率は単調に減少し, ENESはほぼ N に反比例している。また, アーキテクチャ/移民方法に関しては, 成功率, ENES共にUD1が最も良い結果となり, 以下, MS1, MS2, UD2の順となった。同様にして10次元の場合もUD1が最も良い結果となった。

この関数の性質は, D_s の最小値の座標をずらしたものであり, 変数に対する対称性が破れているが, この非対称性が結果に及ぼす影響はそれ程ではないことが, 図3.4-3.7の外形との比較から分かる。

3.4.5 Rastrigin's function に対する結果

図3.20, 3.21は, 5次元Raに対する成功率とENESであり, 図3.22, 3.23は, 10次元Raに対する成功率とENESである。図から分かることは, 成功率は単調減少ではないということである。このように成功率が極大値をとる傾向にあるのは, この関数が多峰性関数であることが関係していると考えられる。また, ENESはほぼ N に反比例している。

アーキテクチャ/移民方法に関しては, 成功率, ENES共にUD1が最も良い結果となった。同様にして10次元の場合もUD1が最も良い結果となった。

図 3.8: 5次元 D_s に対する成功率図 3.9: 5次元 D_s に対する ENES

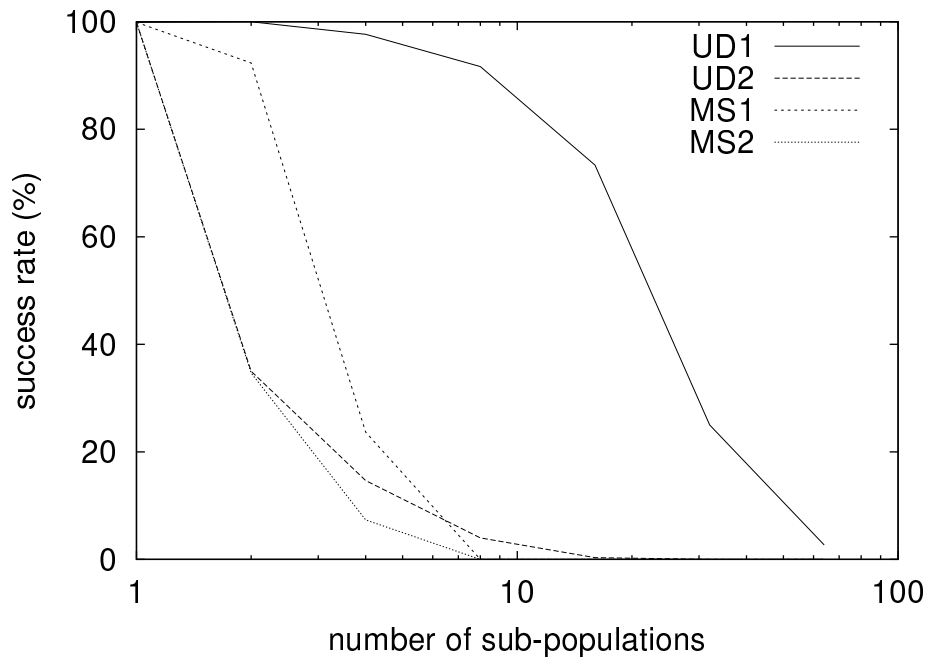


図 3.10: 10次元Dsに対する成功率

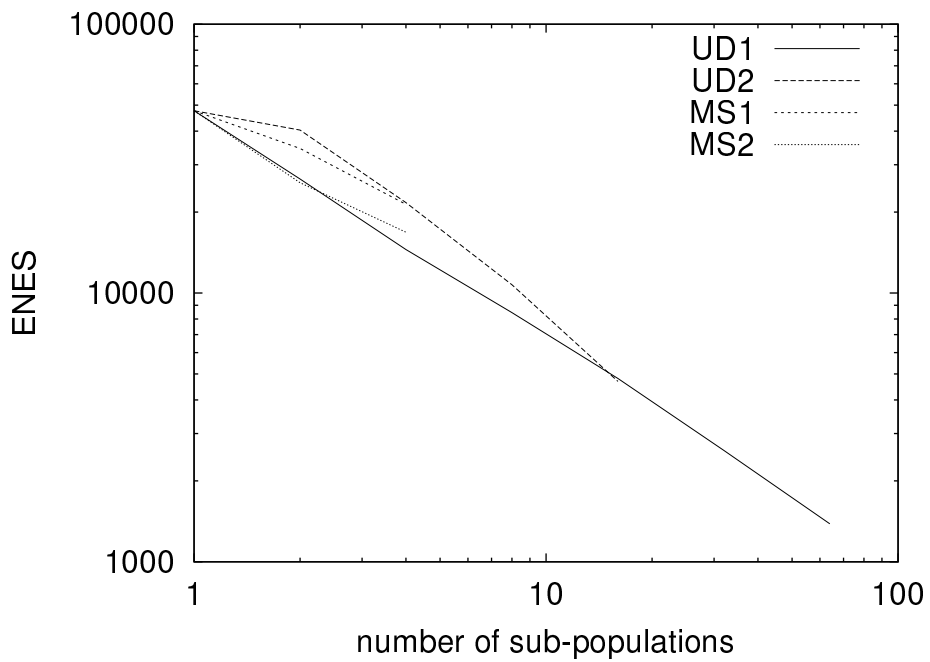
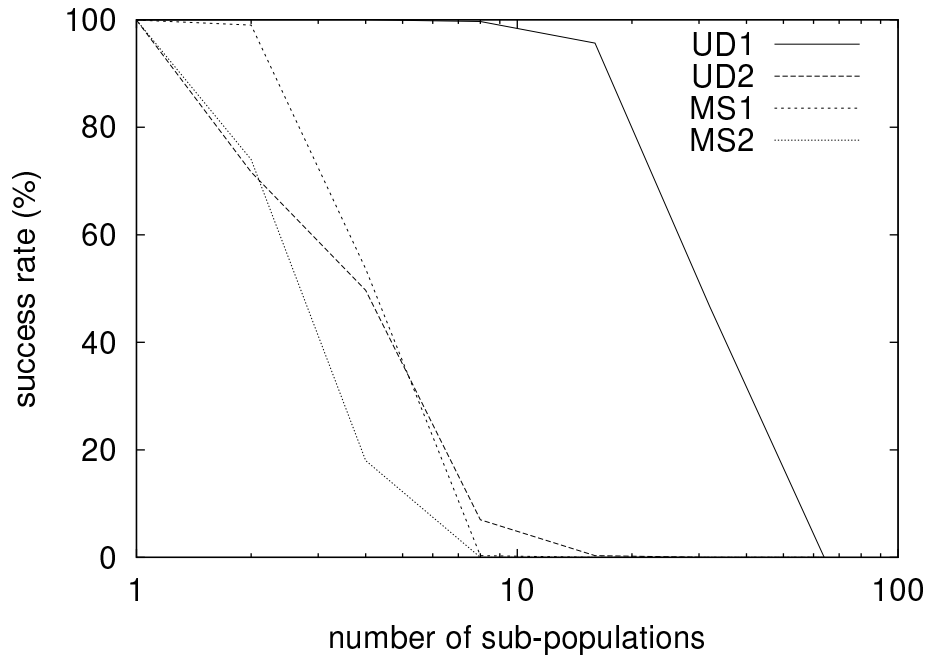
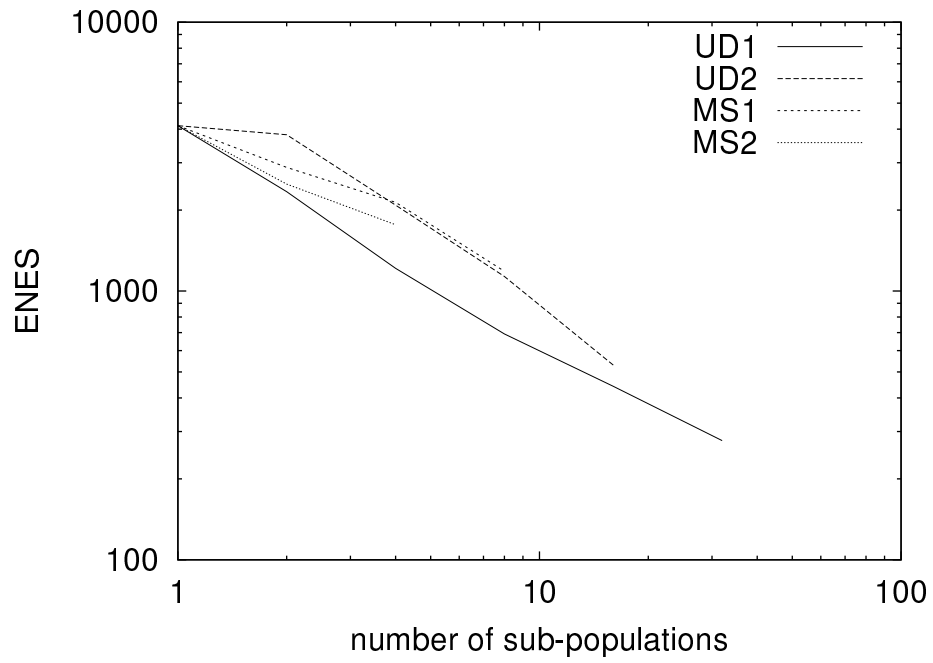
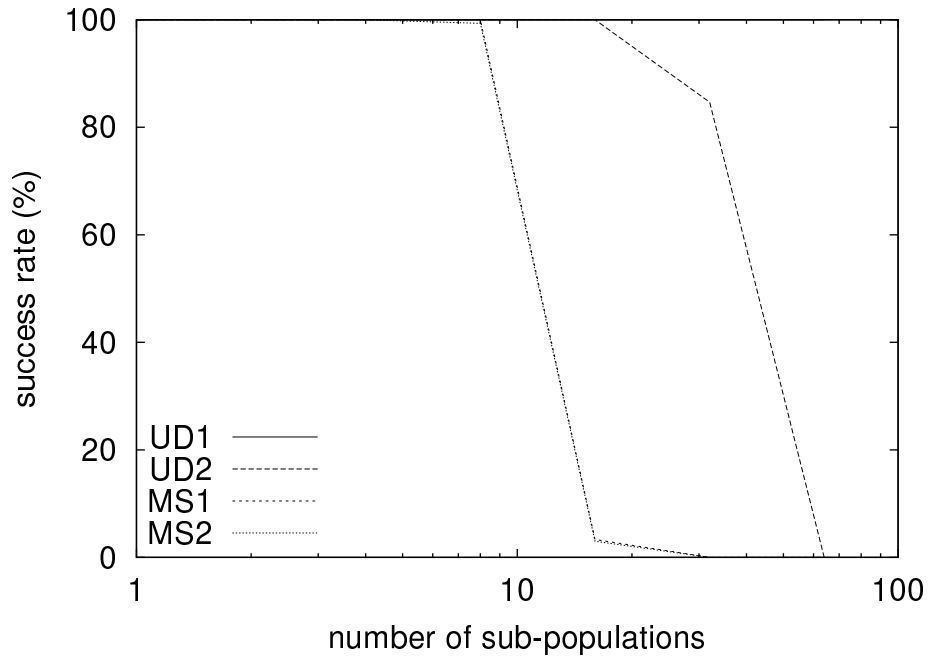
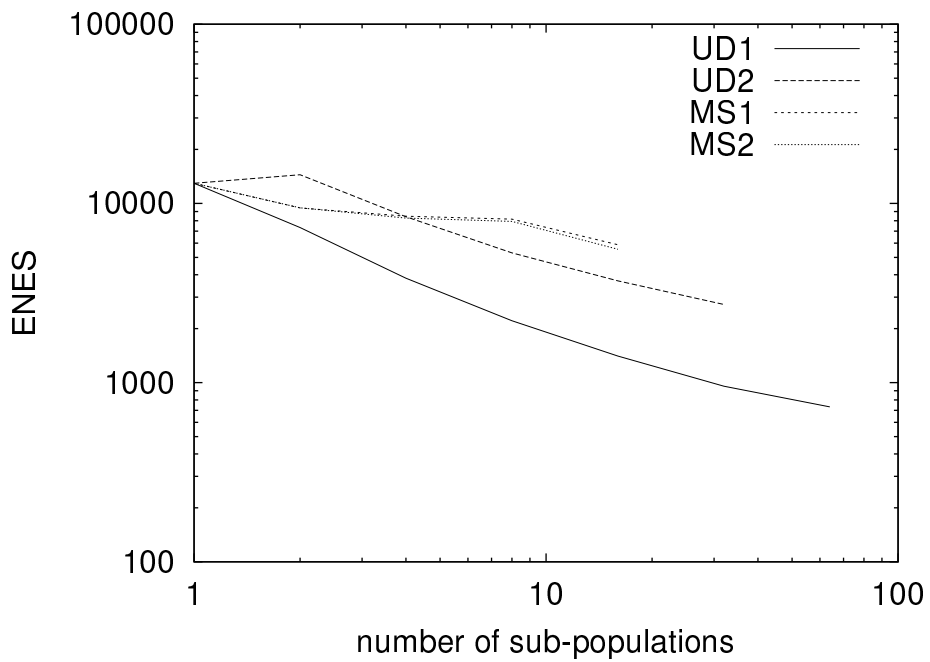


図 3.11: 10次元Dsに対する ENES

図 3.12: 5次元 R_s に対する成功率図 3.13: 5次元 R_s に対する ENES

図 3.14: 10次元 R_s に対する成功率図 3.15: 10次元 R_s に対する ENES

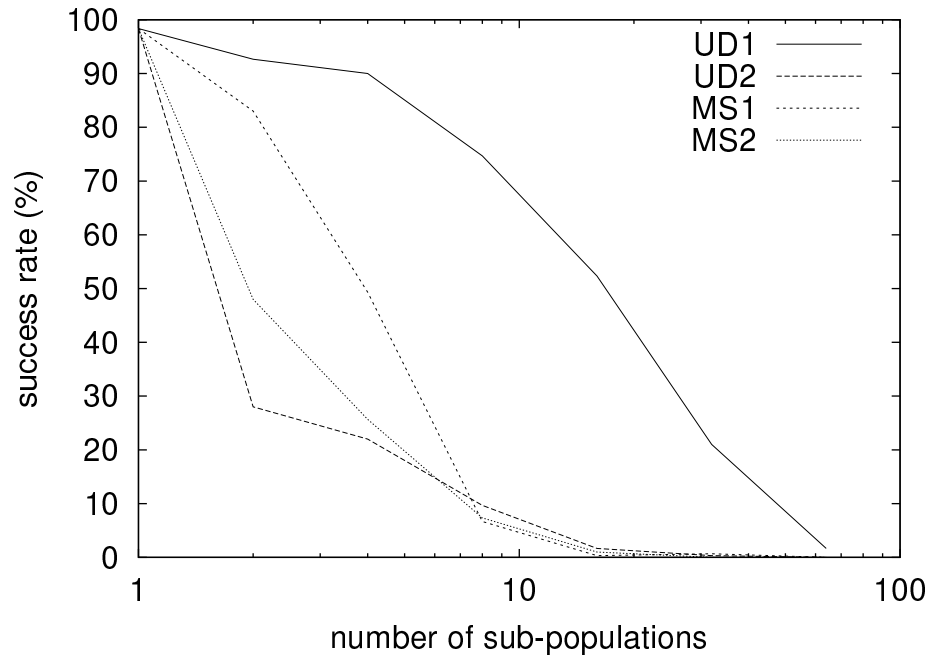


図 3.16: 5次元 Rd に対する成功率

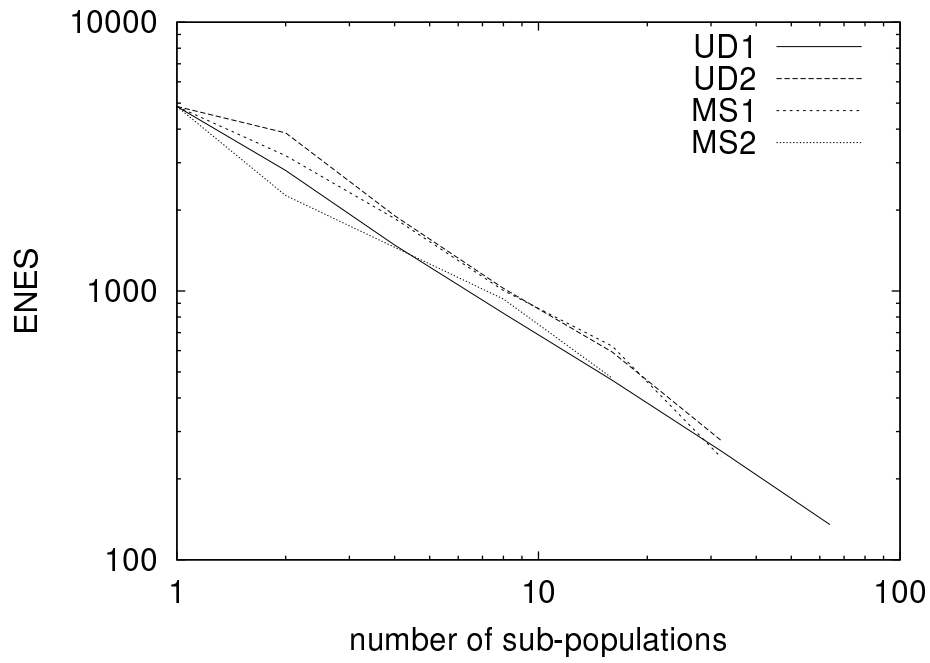


図 3.17: 5次元 Rd に対する ENES

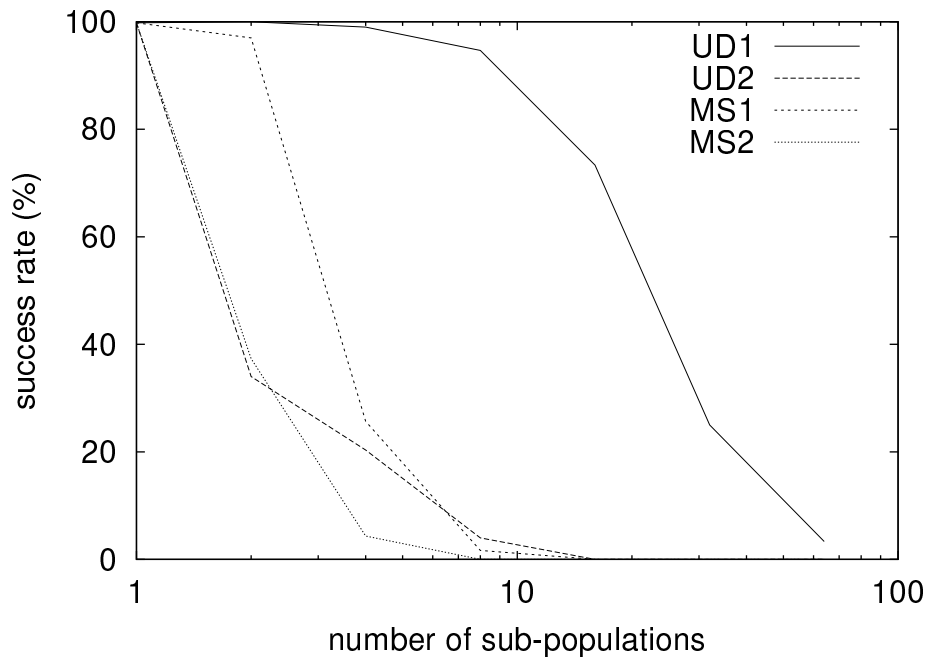


図 3.18: 10 次元 R_d に対する成功率

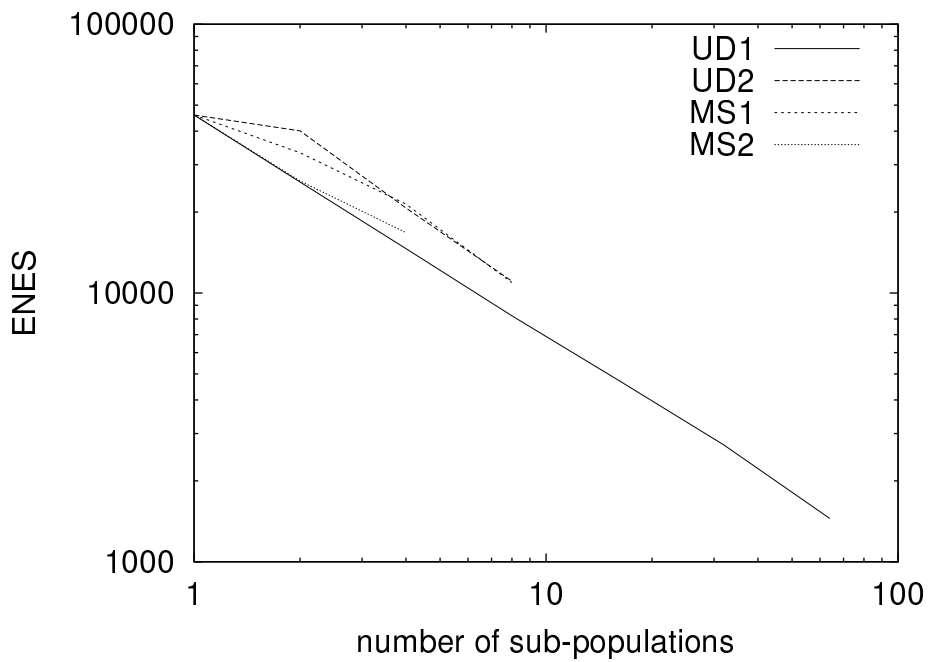


図 3.19: 10 次元 R_d に対する ENES

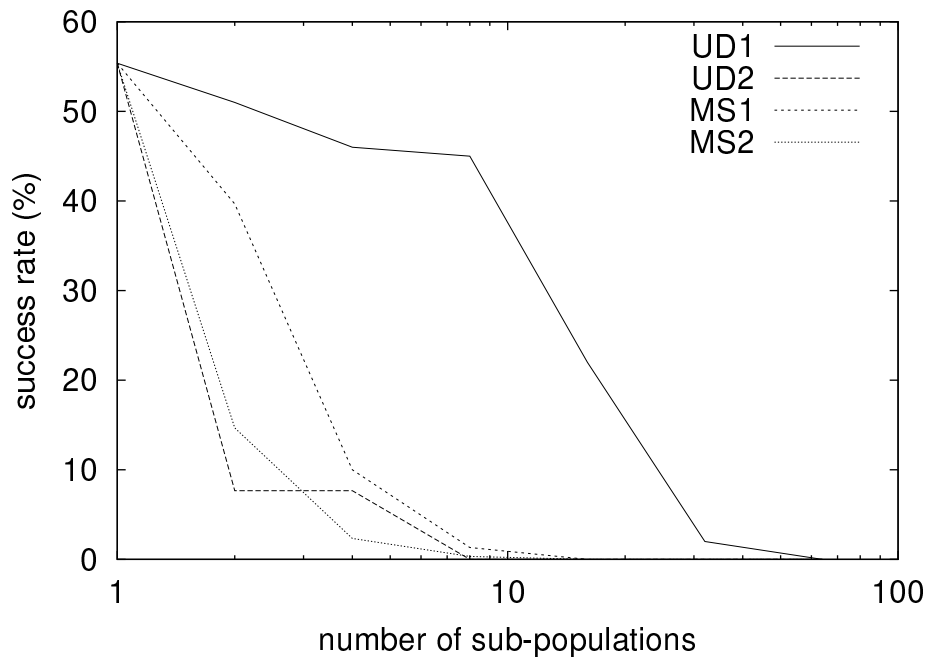


図 3.20: 5次元 Ra に対する成功率

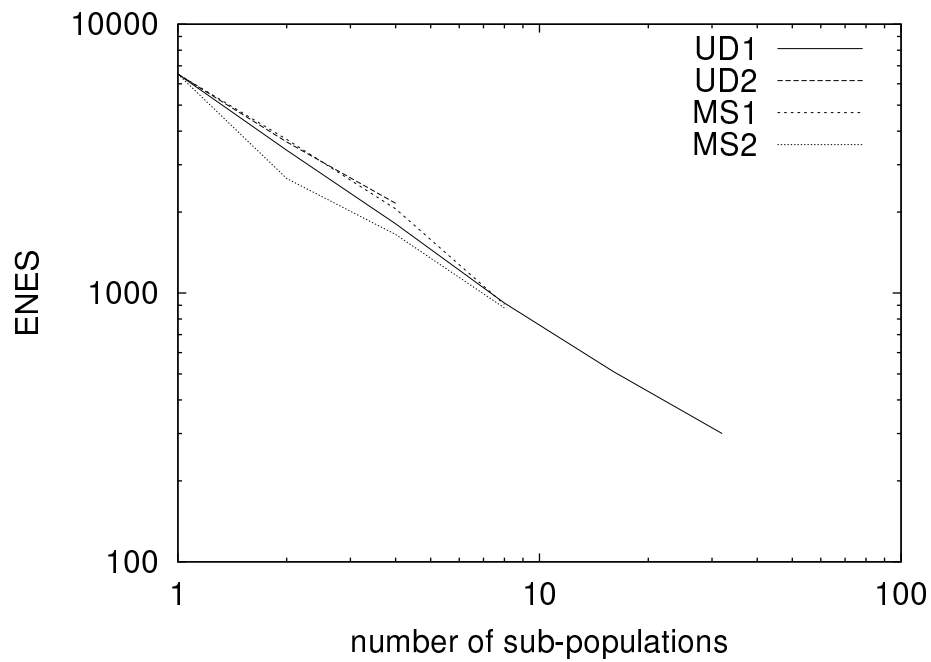


図 3.21: 5次元 Ra に対する ENES

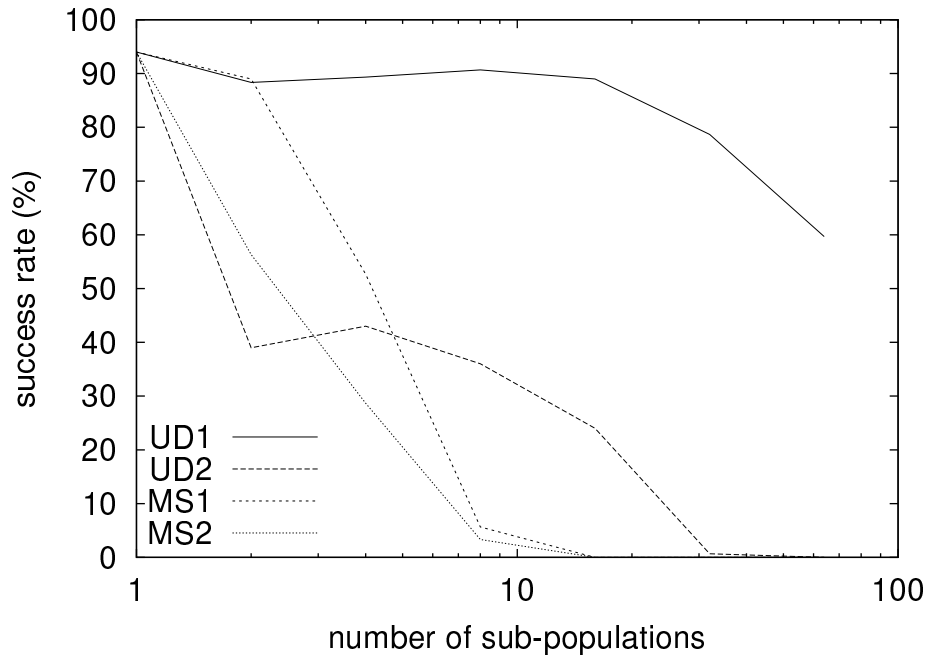


図 3.22: 10次元 R_a に対する成功率

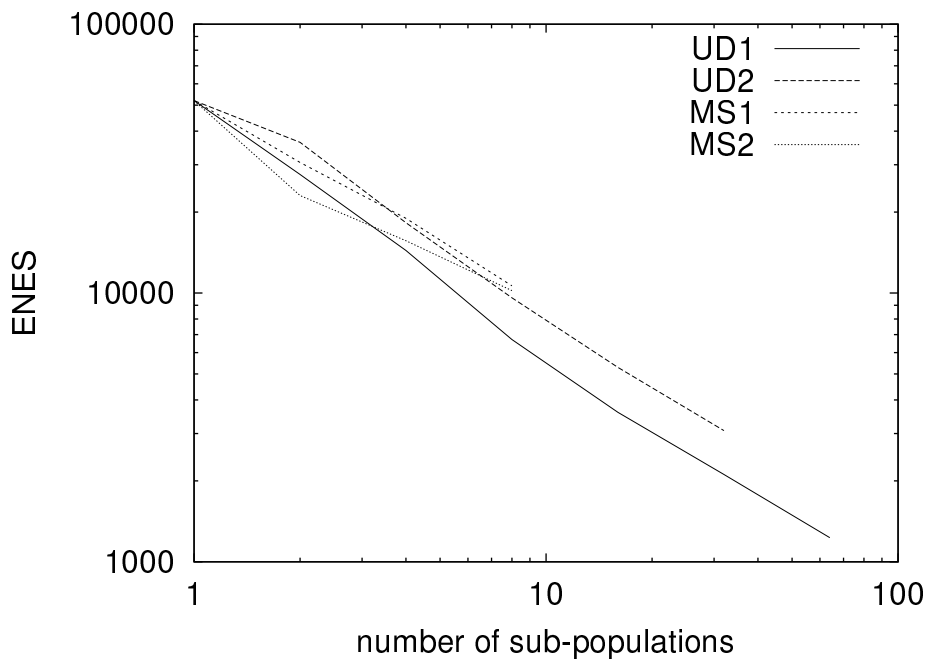


図 3.23: 10次元 R_a に対する ENES

3.4.6 Griewank's function に対する結果

図 3.24, 3.25 は, 5次元 Gr に対する成功率と ENES であり, 図 3.26, 3.27 は, 10次元 Gr に対する成功率と ENES である. 成功率は $N = 1$ のときよりも増加しているケースがある. 5次元では UD1, MS1 であり, 10次元では全てのアーキテクチャで増加している. また, ENES はほぼ $1/N$ で減少している.

アーキテクチャ/移民方法に関しては, 5次元では, UD1, MS1 が優れているようである. 10次元の場合も UD1, MS1 が相対的に優れているようである.

3.4.7 Michalewicz' function に対する結果

図 3.28, 3.29 は, 5次元 Mi に対する成功率と ENES であり, 図 3.30, 3.31 は, 10次元 Mi に対する成功率と ENES である. 成功率は極大値をとるケースがあり, ENES はほぼ $1/N$ で減少している.

アーキテクチャ/移民方法に関しては, 5, 10次元ともに, UD1 が優れているようである.

3.4.8 Shekel's foxholes に対する結果

図 3.32, 3.33 は, 5次元 Sh に対する成功率と ENES であり, 図 3.34, 3.35 は, 10次元 Sh に対する成功率と ENES である. 成功率は, 5次元の場合は全てのケースで極大値をとり, ENES はほぼ $1/N$ で減少している. 10次元の場合は MS1, MS2 で成功率が増加傾向にあり, ENES は UD1 で $1/N$ の振る舞いをしている.

このように, 成功率を一定に保ったまま ENES を $1/N$ にするケースと, ENES をあまり減少させない代わりに成功率を増加させるケースにはっきり別れているのは非常に興味深い現象である.

3.4.9 Langerman's function に対する結果

図 3.36, 3.37 は, 5次元 La に対する成功率と ENES であり, 図 3.38, 3.39 は, 10次元 La に対する成功率と ENES である. この場合も, 全てのアーキテクチャで成功率が極大値をとっているのが分かる. ENES もほぼ $1/N$ の振る舞いであることが確認される.

アーキテクチャ/移民方法に関しては, 5, 10次元ともに, UD1 が優れているようである. 5次元の場合の成功率から, 以下, 順に MS1, MS2, UD2 である.

3.5 実験結果に対する考察と最良戦略

目的関数が単峰性である場合は, 成功率が単調減少し, 多峰性関数に対してはある N で極大値をとる傾向にあることが確認された. この理由は, 単峰性関数では解への収束の速さが重要であるため, 出来るだけ少数個体で探索するのが良く, 多峰性関数では逆に個体の多様性が重要となり, N をある程度大きくする必要があるためである. PfGA 単体 ($N = 1$) で

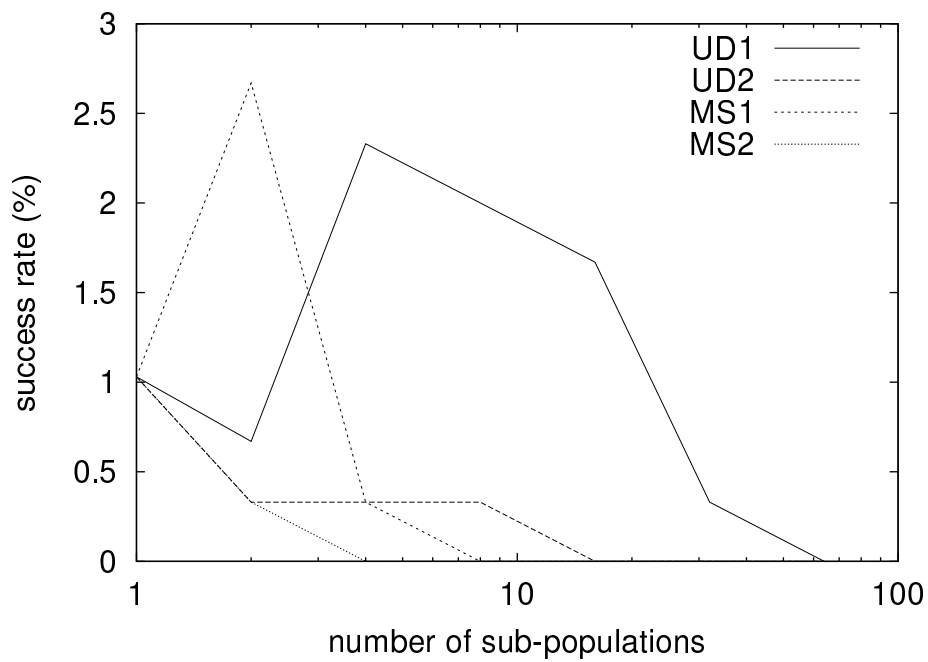


図 3.24: 5次元 G_r に対する成功率

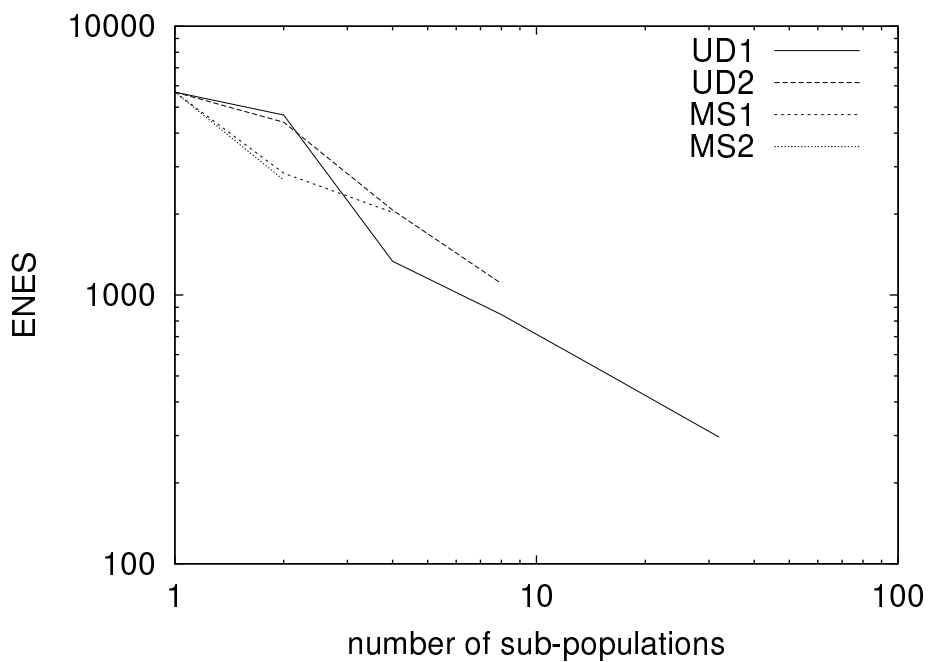


図 3.25: 5次元 G_r に対する ENES

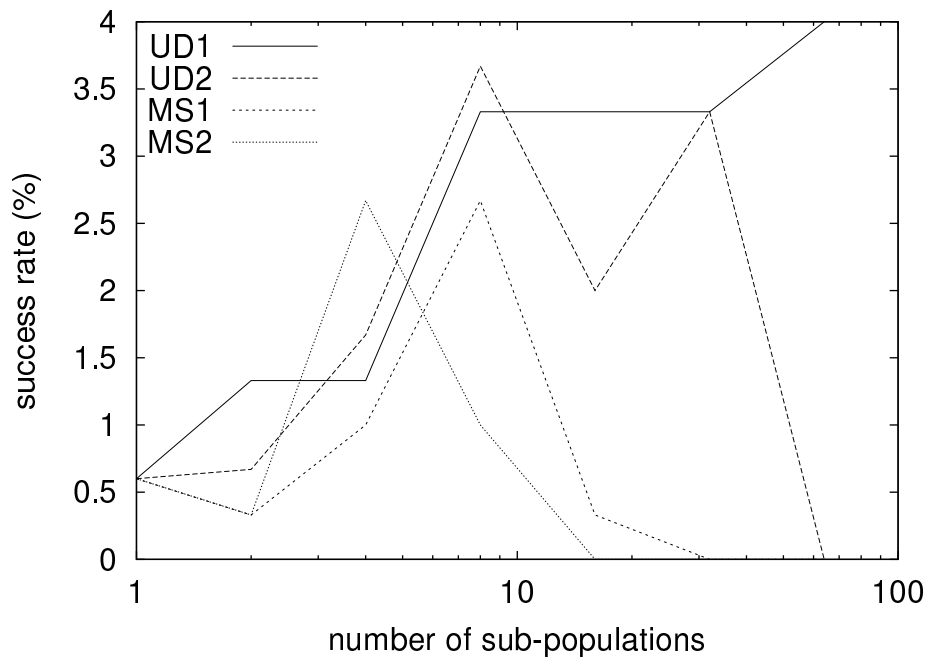


図 3.26: 10次元 Gr に対する成功率

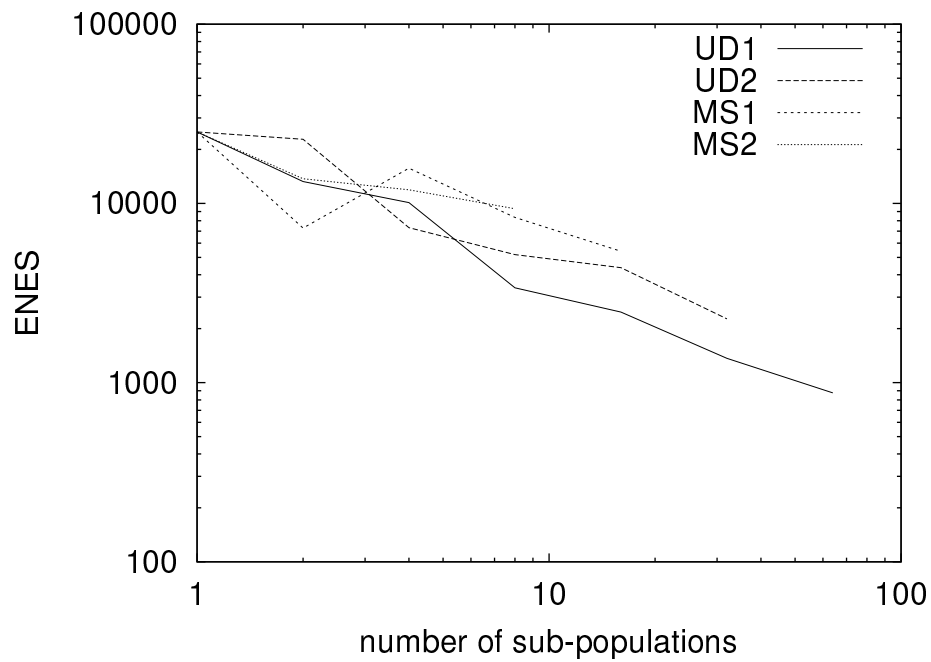


図 3.27: 10次元 Gr に対する ENES

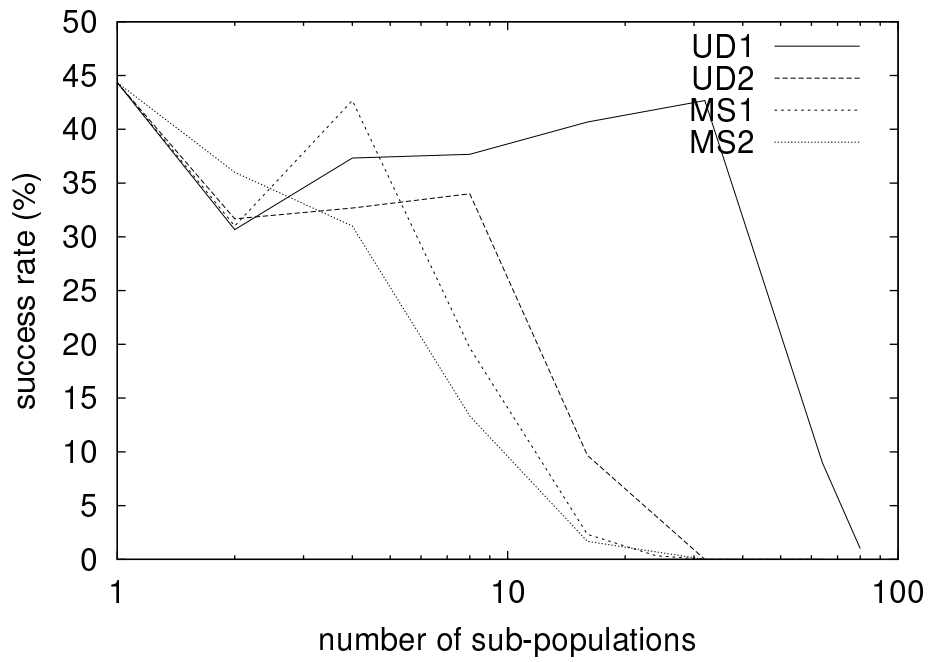


図 3.28: 5次元 M_i に対する成功率

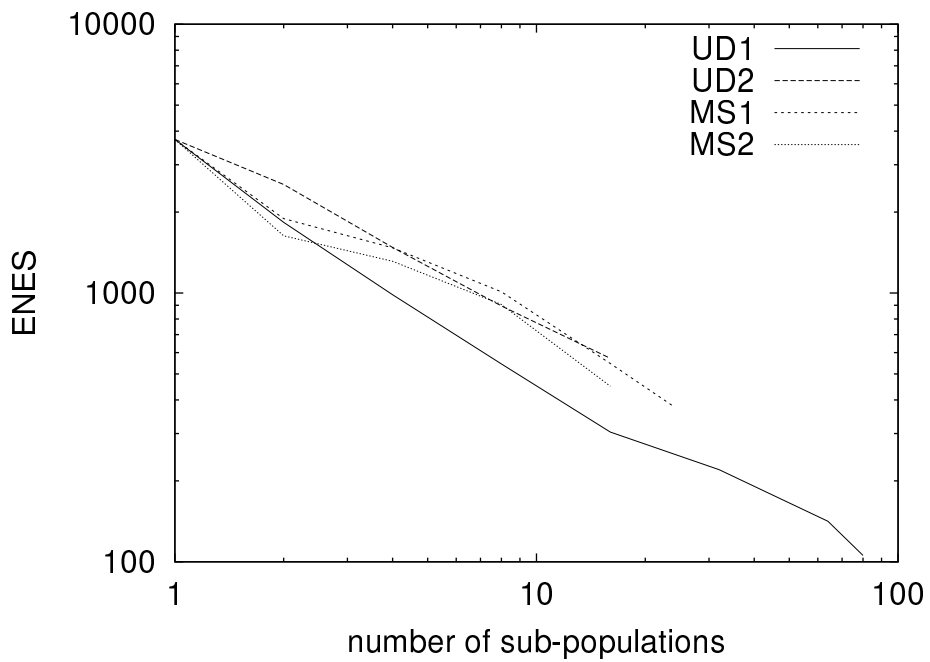
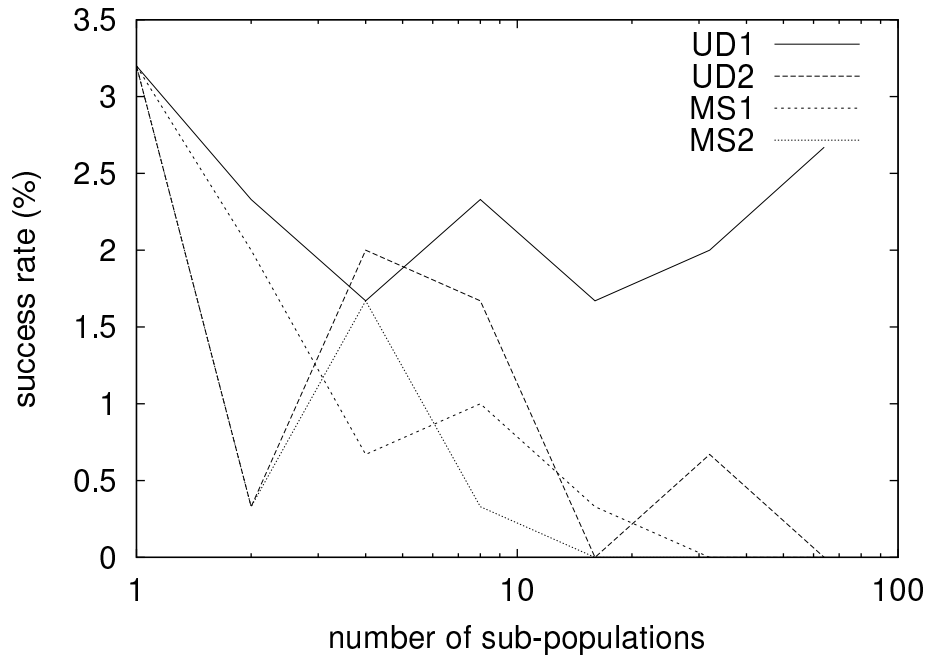
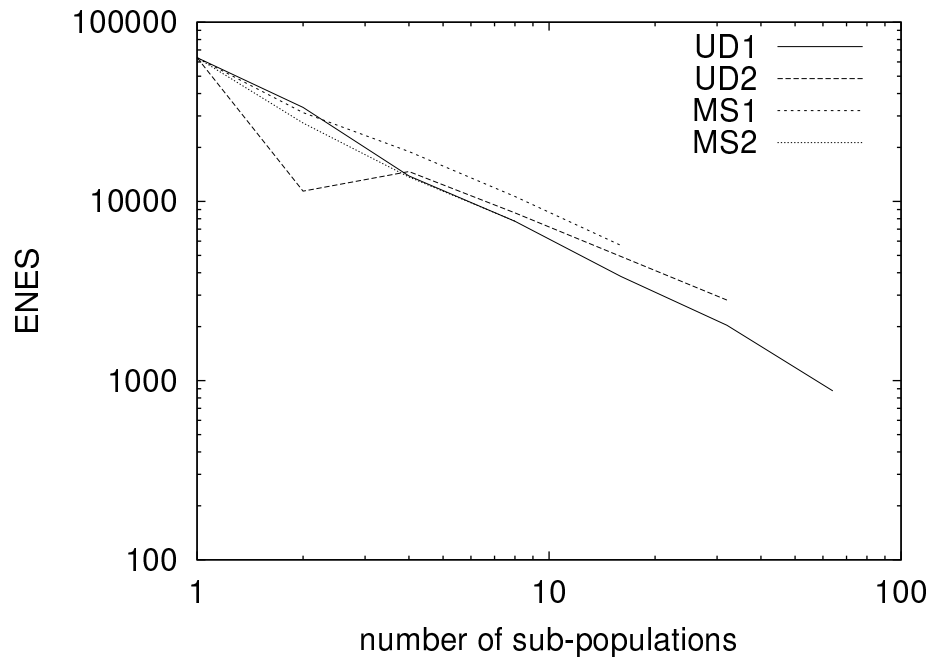


図 3.29: 5次元 M_i に対する ENES

図 3.30: 10次元 M_i に対する成功率図 3.31: 10次元 M_i に対する ENES

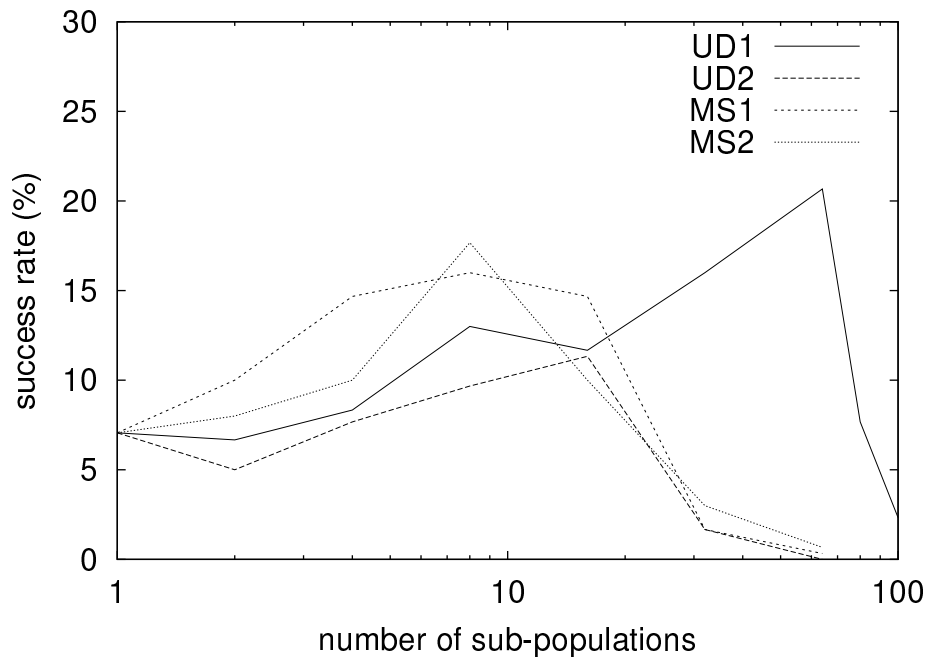


図 3.32: 5次元 Sh に対する成功率

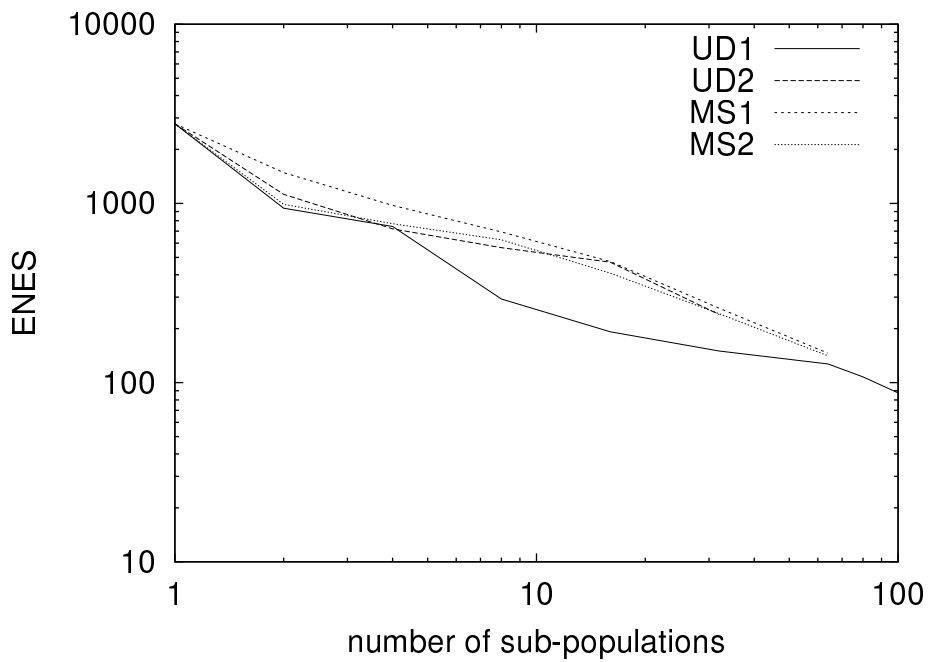


図 3.33: 5次元 Sh に対する ENES

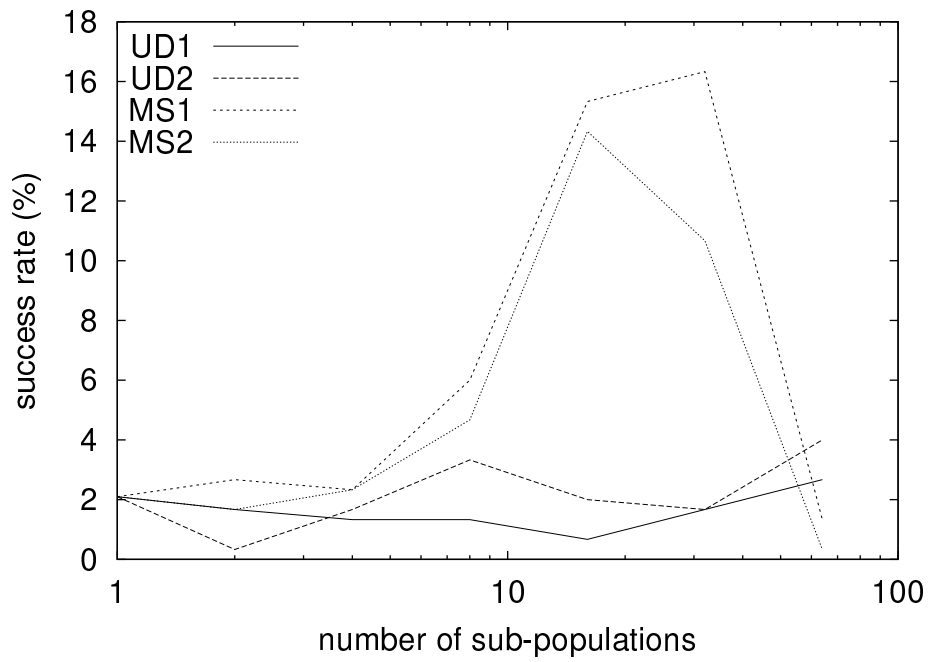


図 3.34: 10次元 Sh に対する成功率

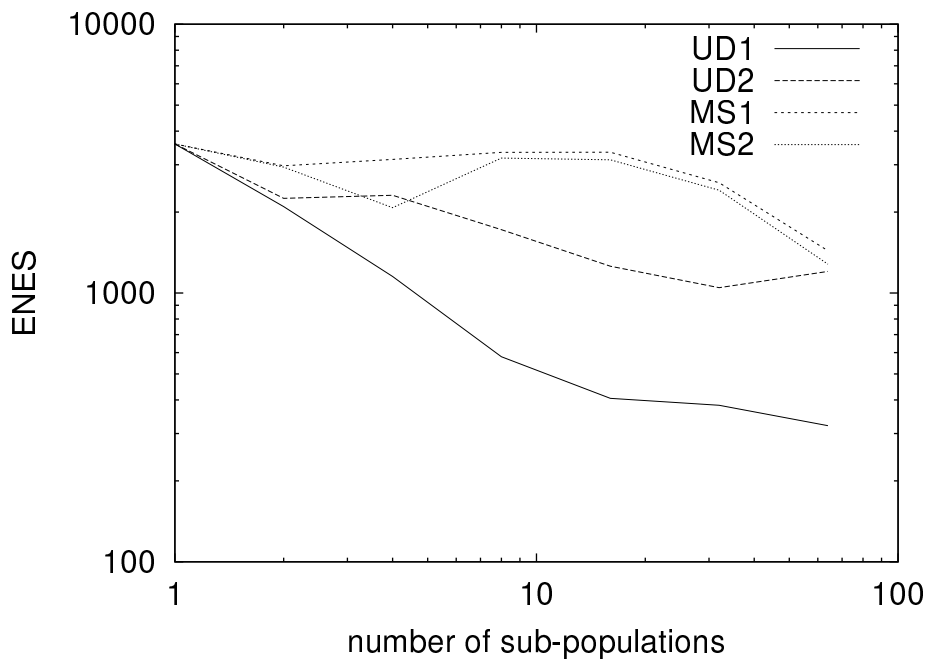


図 3.35: 10次元 Sh に対する ENES

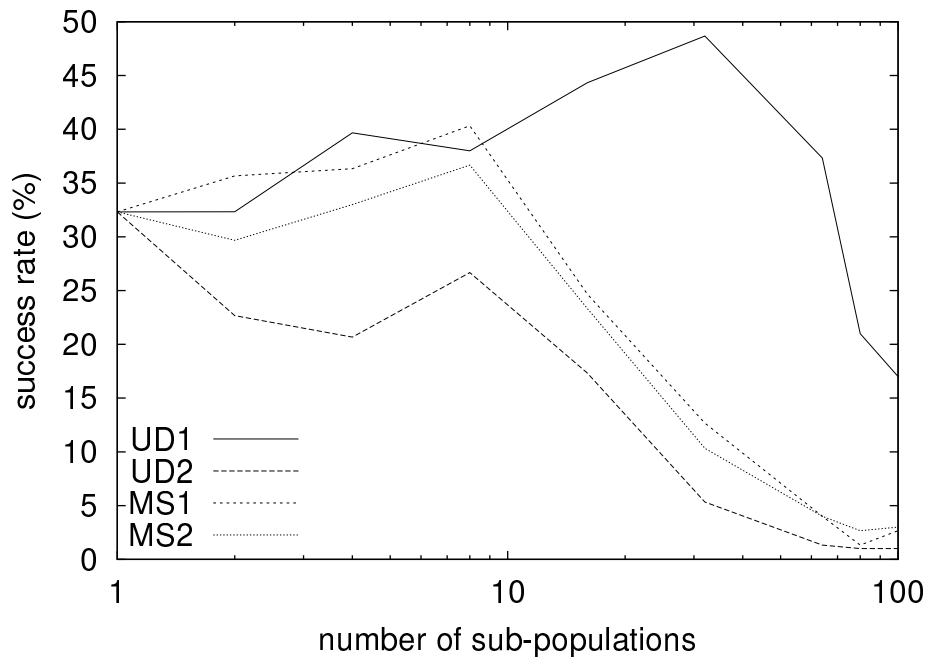


図 3.36: 5次元 L_a に対する成功率

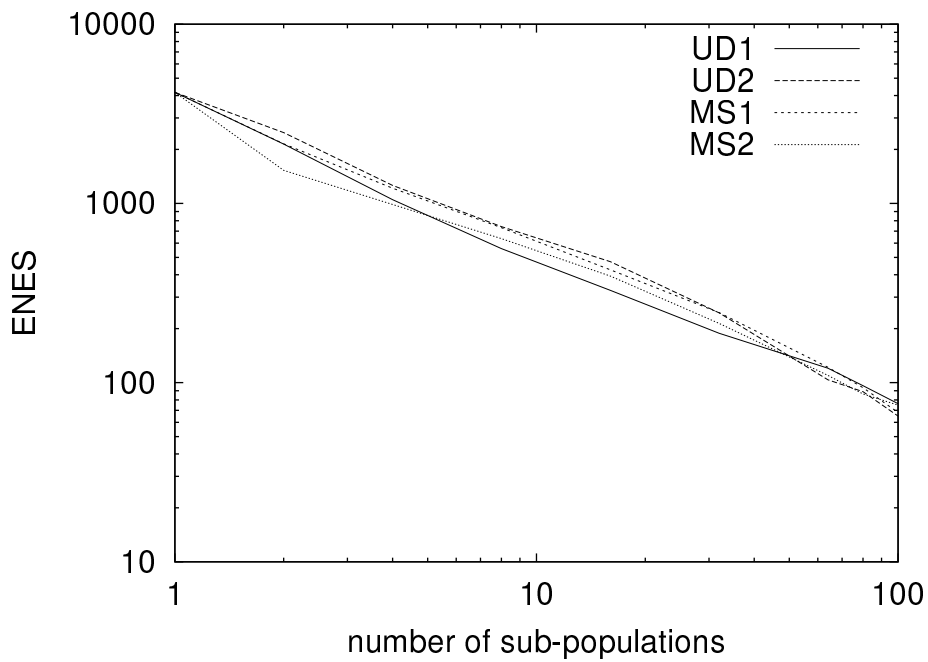
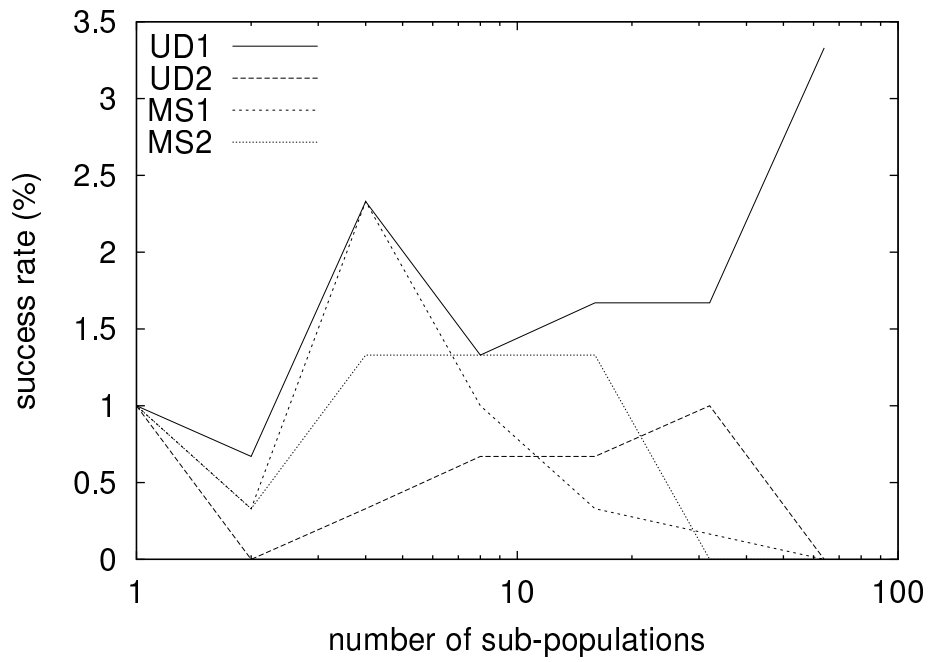
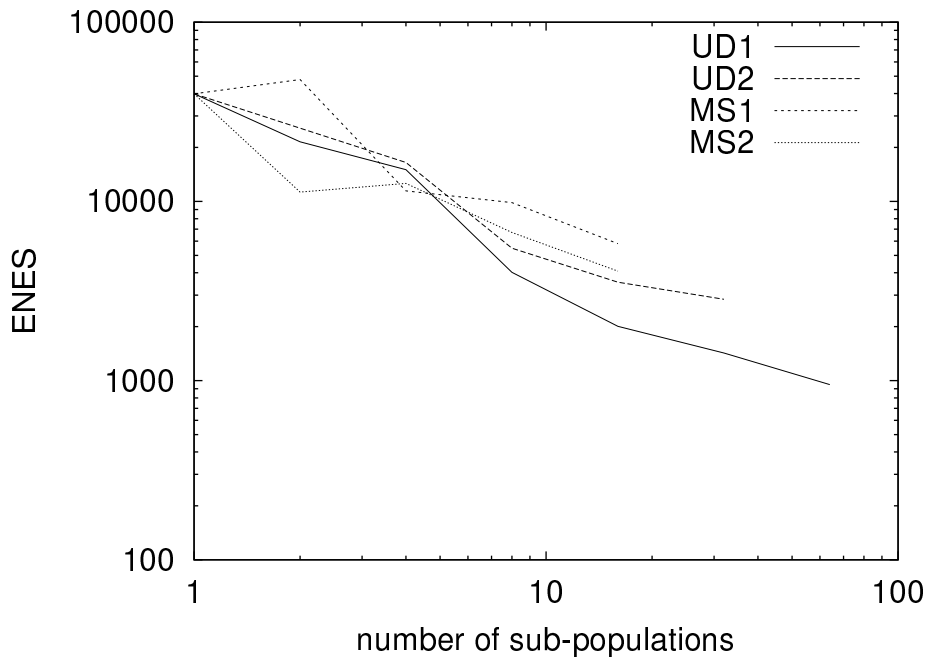


図 3.37: 5次元 L_a に対する ENES

図 3.38: 10次元 L_a に対する成功率図 3.39: 10次元 L_a に対する ENES

の平均個体数は2であるから、多峰性関数では100個以上の個体が探索空間に存在する場合に効率の良い探索が可能となっている。

成功率が極大値をとる N の値は何によって決定されるのであろうか。例えば10次元 Sh に対しては、成功率が $N = 32$ で最大となっていることと、Sh の極小値の数が30個であることを考慮すると、目的関数の極小値の数に依存することが予想される。しかしながらその他の関数に対しては必ずしも当てはまっていないため、さらなる解析が必要である。

また、アーキテクチャ/移民方法に関してはほとんどの場合で UD1 が最も良い結果となり、例外もあるが、以下、順に MS1, MS2, UD2 となった。目的関数の次元、変数の対称性、および変数分離可、不可のような微細な構造の違いに対しても、相対的に UD1 が優れていることが確認された。UD1 が優れているのは、マスタースレーブ型よりも一様分散型のほうが効率が良く、また、移民方法は直接移民する方が効率が良い事を表している。マスタースレーブ型では全局所集団中の最良個体が必ず移民候補となるが、一様分散型では多数の局所的な最良個体が移民候補となっており、この移民候補が多数存在する場合は局所解に陥る可能性は低くなると考えられる。また、移民を階層的に行うのは個体の多様性を増加させ、より効率的に最適解に到達することを期待したためであったが、新たに発生する2個体を交叉・突然変異・評価するための負荷を相殺してまで探索効率を向上させることは出来なかった。

このように、評価回数 ENES と成功率は密接に関係しており、探索効率は多様性と評価回数とのバランスによって決まることがわかる。しかしながら、 N を大きくした場合には、ほとんど全てのアーキテクチャで、 $N = 1$ のとき (PfGA 単体に相当) に比べて ENES が小さくなっている。このことは並列処理自体の効果を端的に示している。

次に、全ての結果を基に、表 3.1 に最良戦略としてまとめた。但し、アーキテクチャ/移民方法については、全て UD1 での結果である。また、ここでいう最良とは、

$$F(N) = \lambda(1 - SR(N)) + (1 - \lambda) \frac{ENES(N)}{ENES(1)} \quad (3.1)$$

が最小となる時の、局所集団数 N に対する成功率 SR と ENES の値を示す。但し、 $\lambda = 0.5$ とする。これは $1 - SR(N)$ (探索が失敗する確率) と規格化された ENES との多目的最適化である。

表 3.1 において、10次元 Sp, 10次元 Rs を除いて、Sp から Rd までは N が比較的小さいところで最良となり、Ra から La までは逆に大きいところで最良となっている。これは、単峰性ならば比較的小さい N (i.e., 個体数小) に対して、一方、多峰性ならば比較的大きい N (i.e., 個体数大) に対してより良い性能が得られることを示している。

3.6 ICEO の他のアルゴリズムとの比較

次に、並列 PfGA の有効性を ICEO'96 に参加した他のアルゴリズム [16] との比較から述べる。表 3.2 はそのうちの上位 5 つまでと、PfGA, 並列 PfGA(UD1) の結果を示したものである。各欄は、上から順に ENES, BV(Best Value), RT(Relative Time) の値である (第2章を参照)。また PfGA は $N = 1$ での結果に相当する。UD1 の欄の (N) は ENES が最小となった時の局所集団数を表している。ENES の大小で比較すると、並列 PfGA では関数ごと

表 3.1: 並列 PfGA(UD1) の最良戦略

func	dim	uni sym sepa	N	succ(%)	ENES
Sp	5	○ ○ ○	16	95	429
	10		64	100	667
Ds	5	○ ○ ×	8	74	816
	10		8	92	8,451
Rs	5	○ × ○	16	96	442
	10		64	100	733
Rd	5	○ × ×	4	90	1,482
	10		8	95	8,227
Ra	5	× ○ ○	8	45	918
	10		16	89	3,598
Gr	5	× ○ ×	32	0.3	296
	10		64	4	877
Mi	5	× × ○	32	43	220
	10		64	3	877
Sh	5	× × ×	64	20	127
	10		64	3	321
La	5	× × ×	32	49	189
	10		64	3	950

に順位は異なり, UD1 では, Michalewicz 関数, Langerman 関数で 1 位, 他の関数では 2 位となっており, UD1 の探索能力の高さが分かる. BV に関しては, 並列 PfGA は PfGA に比べて若干精度が落ちているようである. また RT については, N が大きくなるほど RT も大きくなった. RT の定義から, N が大きくなると移民などの通信に要する負荷が増加し, (CT-ET) の部分が相対的に大きくなることが原因と考えられる.

3.7 まとめ

本章では PfGA の拡張として, 並列分散処理アーキテクチャにおける移民選択法の効果について述べた. 2 種類の並列分散アーキテクチャと 2 種類の移民方法を組合せ, 計 4 つの方法を並列コンピュータ上に実装し, 同期的に実行して性能を比較検討した. ベンチマーク問題として 9 関数の 5 次元版と 10 次元版を用いて探索性能を評価した結果, 局所集団数を増加させることにより, 全ての関数に対して移民の効果が現れた. また, 全評価回数の総和を固定したにもかかわらず, 並列の効果によって成功率を上昇させることができた.

また, 4 つのアーキテクチャ/移民法では, UD1, MS1, MS2, UD2 の順に性能が良いことが判明した. 局所集団数を増加していけば移民の効果によって, 逐次処理に比較して成功率も向上することが確認できた. この移民効果は全ての型で見られたが, 特に UD1 で顕著と

表 3.2: 第1回ICEO と PfGA, 並列 PfGA の比較. (N) は並列度

ENES BV RT	Bi-Pa [17]	Li [18]	Sto-Pri [19]	Van Ke [20]	Se-Be [21]	PfGA	PPfGA UD1 (N)
Sp5	20	243	736	1,452	326	4,067	173 (56)
	3.88e-15	0.0	-	1.5e-30	-	0.0	4.52e-7
	2	12.7	4.67	75	0.35	0.91	2.30
Gr5	41	21,141	5,765	22,039	35,637	6,785	846 (8)
	7.99e-6	1.69e-5	-	3.2e-12	-	4.66e-7	4.22e-5
	2	3.1	1.79	2	0.35	0.90	1.88
Sh5	74	6,318	76,210	51,845	13,836	1,619	87 (100)
	-10.327	-10.403	-	-9.83	-	-10.40	-9.7624
	2	0.25	0.80	0.96	0.35	0.33	2.24
Mi5	120	6,804	1,877	10,661	9,925	5,131	105 (80)
	-4.6876	-4.687	-	-4.687	-	-4.688	-4.68749
	2	1.28	1.11	3.9	0.35	0.90	1.93
La5	176	4,131	5,308	11,343	74,720	5,274	76 (100)
	-1.499	-1.499	-	-1.49	-	-1.499	-1.49752
	2	1.62	1.35	1.8	0.35	0.43	1.93

なった。これは、UD1が最も単純な構造を持っているためである。これに対し、MS1は単純な構造であるが、初期収束などの観点から、UD1に比べてそれほど優秀ではなかったということである。

国際コンテスト ICEO との比較においても、並列 PfGA はその他のアルゴリズムに比べて遜色のない優れたアルゴリズムであることが示された。

ところで、並列計算にはその並列度というパラメータが常に付随するが、これをあらかじめ設定しないという拡張も考えられる。すなわち、並列 PfGA における局所集団数が目的関数に応じて動的に決定されるようなルール作りが今後の課題として挙げられる。

第4章 遺伝子重複説に基づいた遺伝的アルゴリズムの提案と性能評価

4.1 概要

本章では、探索における成功率を向上させる目的で、遺伝子重複説に基づいたGAの提案と評価を行う[31]。成功率を向上させるためには、GAにおける個体集団の多様性の増加が最も妥当な方法であるが、単に個体数を増加させるだけならば、これはパラメータの調節の問題となってしまう。また、個体数を増加することによって個体の評価回数も増加してしまい、与えられた時間内に探索できなくなる可能性もある。そこで、優秀なスキーマを保持し、それらを重複していくことによって多様性を増加させる方法を考える。

大野乾が1970年代に提案した遺伝子重複説[32]は、ウイルス、植物、動物などを含めたあらゆる生物の進化の過程では、それまでの蓄積されてきた遺伝子の断片をコピーしたり再利用することにより、より高次の生物への進化を促進させる効果があると主張している。これを大野は、「一創造、百盗作」という簡潔な言葉で表現している[32]。この遺伝子重複は、同一染色体の染色分体間の不等交換、減数分裂過程での相同染色体間の不等交叉、DNAの部分的な繰り返し複製等によって起こるとしている。これらの遺伝子重複機構からヒントを得て、優秀なスキーマを保持するアルゴリズムを考案する。

遺伝子重複は遺伝子長の異なる個体に対しても適用できる遺伝的オペレータであり、多次元の関数最適化問題を解く場合に、その効果を発揮できるものである。その適用方法は、まず変数を部分次元毎に遺伝子にコーディングし、部分空間における適応度関数を設定し、GAを働かせることにより(準)最適解を求める。この(準)最適解に対応する遺伝子を持つ個体を重複することにより、より高次元の空間における最適化問題を解く。この求解プロセスを、異なる遺伝子長を持つ個体が局所集団間を移民することにより実現する[31]。重複方法としては、4つの遺伝子重複モデル(連鎖型、伸長型、結合型、拡張結合型)を考える。

また、局所集団を進化させるベースとなるGAとしては、SSGAとPfGAの2つを比較検討する。SSGAは多数個体で解空間を探索するのに対し、PfGAは少数個体で解空間を探索する。これらを比較する意味は、遺伝子重複の効果が、そのベースとなるアルゴリズムに依存するかどうかを明確にするためである。

4.2 遺伝子重複アルゴリズム

生態系では一般的に、長い遺伝子を持つ高等生物ほど優れているとされている。遺伝子重複をGAに実装する従来研究[33]においても、長い遺伝子ほど適応度が高くなるように設定されているため、長い遺伝子を持つ個体が生き残りやすくなるのは当然である。しかしな

がら、GAを最適化の手法として用いることを前提とすると、目的関数が必ずしもそうになっているとは限らず、また、ここでの目的である、多様性の増加や優れたスキーマの保持という点ともあまり関連がない。

そこで、関数の各次元毎の遺伝子に対応する部分空間で進化させ、それらを適当なタイミングで連結するという方法を考える。最も一般的で簡単であろうと考えられるのは以下の4種類の重複方法である。

- タイプA (遺伝子連鎖型): 各次元毎の遺伝子を持つ個体を一挙に連結する。
- タイプB (遺伝子伸長型): 繰り返しによって、より長い遺伝子を作る。
- タイプC (遺伝子結合型): 長さ i の遺伝子と長さ j の遺伝子を連結して、長さ $i+j$ の遺伝子を作る。これを i と j の全ての組み合わせで行う。
- タイプD (遺伝子拡張結合型): タイプCが遺伝子座を考慮していないのに対し、遺伝子座を考慮してタイプCを拡張したものである。

以上の4つのタイプにおいて、長さ i の遺伝子を持つ個体はSSGAまたはPfGAにより進化させ、適当なタイミングで連結、伸長、または結合させる。これにより長さ k の個体が生じたとすると、それに応じた集団に”移民”させる。移民先では移民個体数だけ個体数が増加するので、悪い方から同数だけ淘汰して、集団サイズを元に戻す。

このように、個体の遺伝子長が固定ではなく可変であり、短い遺伝子が連結(重複)することにより、より長い遺伝子を持つ個体が生じる構造を持っている。また、異なる長さを持つ個体を共存させるのではなく、あくまでも別々の局所集団で進化させるという点が従来法とは異なる特徴となっている。

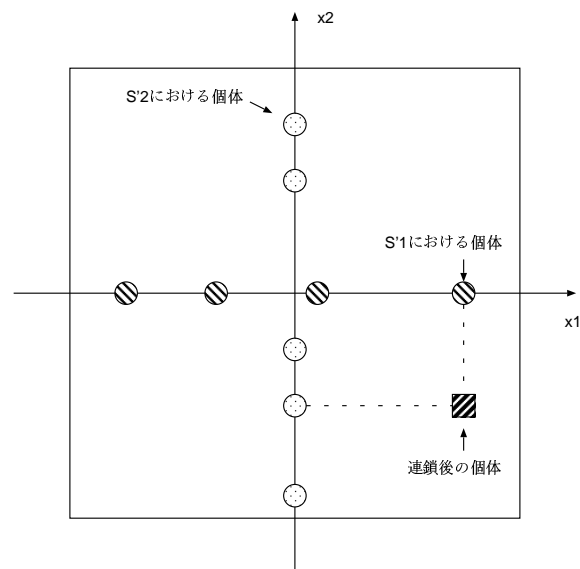


図 4.1: タイプ A における遺伝子重複のイメージ

なお、注意点としては、短い遺伝子の適応度についてであるが、例として、タイプ A の場合の探索イメージを図 4.1 に示す。この例では、2 次元平面上の x_1 軸、および x_2 軸がそれぞれ局所探索空間であり、連鎖後の個体は 2 次元平面全体が探索空間である。このようにする理由は、多次元の目的関数に対する事前知識が得られない場合をも想定しているからである。すなわち、 N 次元関数の入力変数が N 個、出力が 1 個となるような、いわばブラックボックスを扱うような場合に対しても適用できるアルゴリズムを目指しているからである。

また、この例のように、1 次元空間では、どの直線を用いるかについては任意性があり、さまざまなバリエーションが可能であるが、ここでは簡単のため x_i 軸上とした。もちろん、これにより、目的関数に変数分離可能であれば、部分問題に分割して解くことになり、容易に解けることが期待される。また、変数分離不可能な関数に対してはどれほどの効果があるかは未知であるが、個体の多様性を増加させることは明らかであるため、期待に値するものである。

次に図を用いながら、各タイプの詳細について説明する。

4.2.1 タイプ A (遺伝子連鎖型)

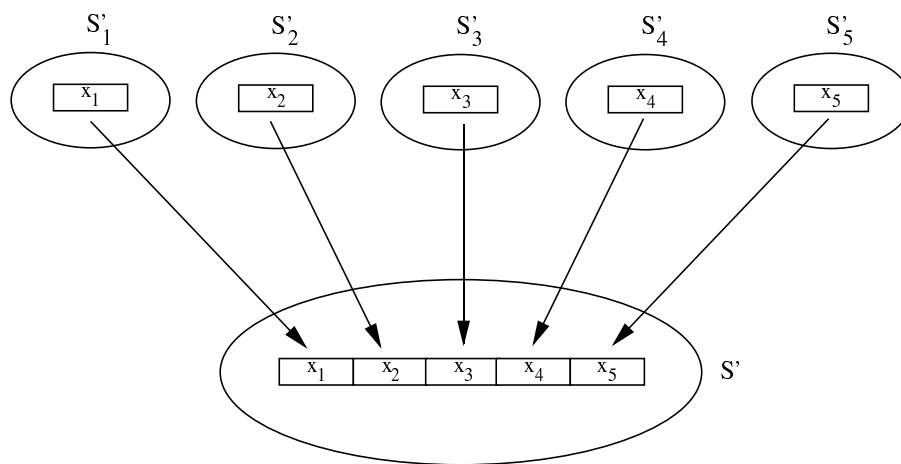


図 4.2: $N = 5$ の時のタイプ A (遺伝子連鎖型) における遺伝子重複法

問題を部分問題に分割して解く方法のうち、この連鎖型が最も簡単な方法である。まず、 N 次元の目的関数 $f(x_1, x_2, \dots, x_N)$ に対し、1 変数 x_i ($i = 1, 2, \dots, N$) を l bit に Gray coding した遺伝子 $[x_i]$ を発生させ、局所集団 S'_i の初期個体とし、進化させる。ただし、適応度の評価は任意性があるが、最も簡単な方法は、 $f(0, \dots, 0, x_i, 0, \dots, 0)$ により評価するものである。また、 N 個の全変数を Nl bit に Gray coding した遺伝子 $[[x_1] \cdots [x_N]]$ を発生させ、 S' の初期個体とし、同様に進化させる。適応度の評価は $f(x_1, x_2, \dots, x_N)$ を用いる。よって局所集団数は $N + 1$ 個である。各局所集団 S'_i ($i = 1, \dots, N$)、及び S' における一定世代の進化が終了した時点で、 S'_1, S'_2, \dots, S'_N から最良子個体を選び、遺伝子配列を結合させて新しい個体 $[[x_1] \cdots [x_N]]$ を生成し、移民レート R で S' へ移民させる (PfGA ベースの場合)。また

は上位 P 個を S' へ移民させる (SSGA ベースの場合, 以下のタイプ B~D についても同様). また, S' では個体数を元に戻すため同数の最悪個体を淘汰する (図 4.2 参照).

4.2.2 タイプ B (遺伝子伸長型)

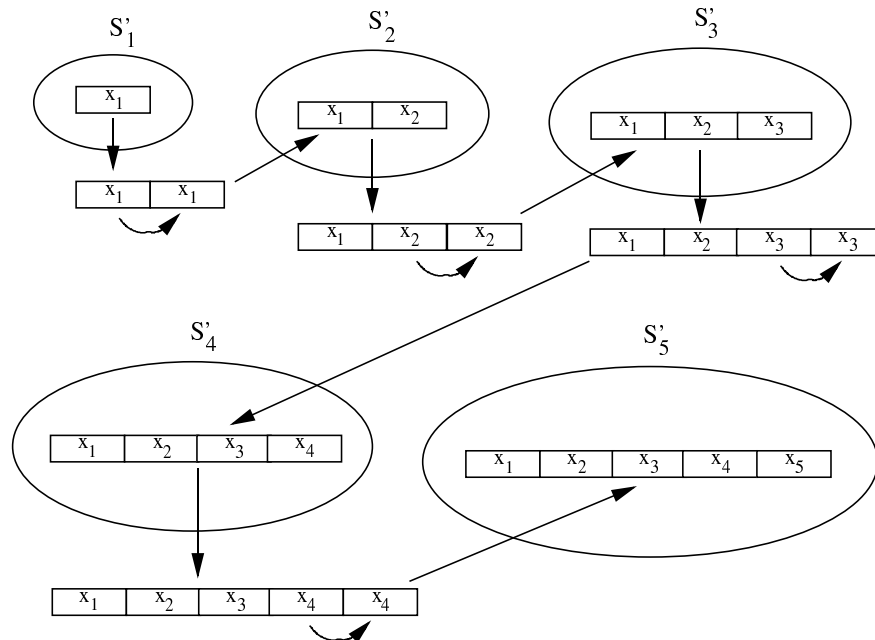
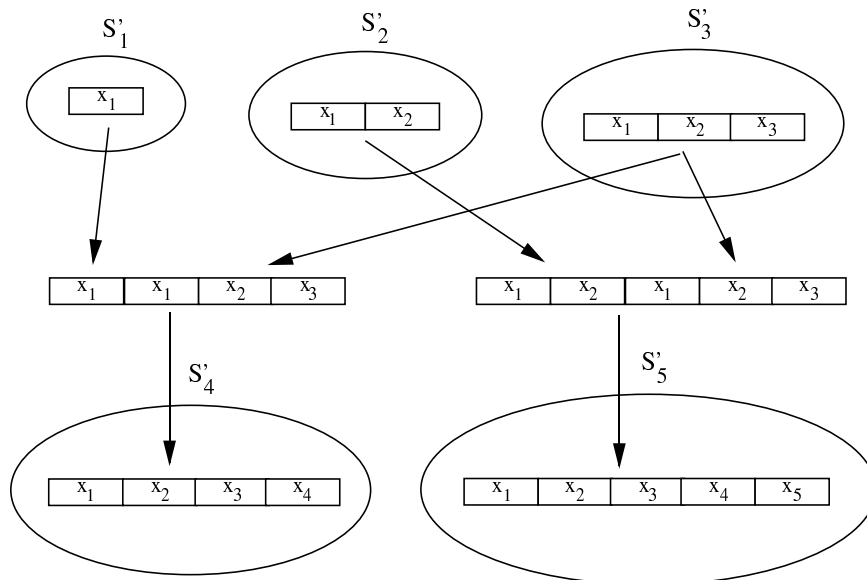
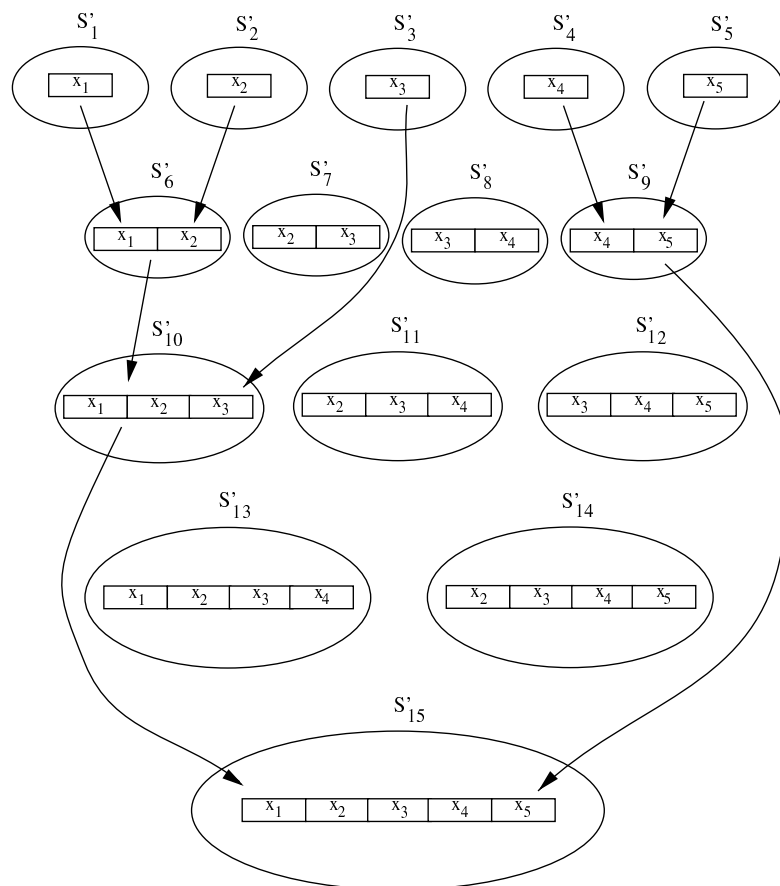


図 4.3: $N = 5$ の時のタイプ B(遺伝子伸長型)における遺伝子重複法

i 個の変数列 x_1, x_2, \dots, x_i を $i \times l$ bit に Gray coding した遺伝子 $[[x_1][x_2] \cdots [x_i]]$ を発生させ, 局所集団 S'_i の初期個体とし, 進化させる. ただし適応度の評価は $f(x_1, x_2, \dots, x_i, 0, \dots, 0)$ で計算する. 局所集団数は N 個である. 各局所集団 S'_i の進化が一定世代終了した時点で, S'_i における最良子個体から遺伝子 $[[x_1][x_2] \cdots [x_i][x_{i+1}]]$ を生成する. ただし, $[x_{i+1}] = [x_i]$ である. この新しい個体を移民レート R で S'_{i+1} へ移民させ, S'_{i+1} では最悪個体を淘汰する (図 4.3 参照).

4.2.3 タイプ C (遺伝子結合型)

Type B に対し, 移民方法を変更したものを考える. 各局所集団の進化が一定世代終了した時点で, S'_i と S'_j における最良子個体を組合せ, S'_{i+j} ($i+j \leq N$) へ移民レート R で移民させる. S'_{i+j} では最悪個体を淘汰する (図 4.4 参照). 局所集団数は N 個である.

図 4.4: $N = 5$ の時のタイプ C(遺伝子結合型) における遺伝子重複法図 4.5: $N = 5$ の時のタイプ D(遺伝子拡張結合型) における遺伝子重複法

4.2.4 タイプ D (遺伝子拡張結合型)

図 4.5 に示すように, 1 変数 x_i ($i = 1, 2, \dots, N$) を Gray coding した遺伝子 $[x_i]$ を発生させ, 局所集団 S'_i の初期個体とする. 次に, 連続する 2 変数 x_i, x_{i+1} を Gray coding した遺伝子 $[x_i][x_{i+1}]$ を発生させ, 局所集団 S'_i ($i = N + 1, N + 2, \dots, 2N - 1$) の初期個体とする. ただしこれらの集団では適応度は $f(0, \dots, 0, x_i, x_{i+1}, 0, \dots, 0)$ で求めるものとする. 同様にし, 連続する変数の数を増やしていき, N 変数までの遺伝子を発生させ, 合計 $N(N + 1)/2$ 個の局所集団を生成し, 進化させる. 各局所集団 S'_i における一定世代の進化が終了した時点で, 移民レート R で S'_i と S'_j の最良子個体を結合し, S'_k への移民を行う. ただしこのとき, 遺伝子結合の条件としては, 2 つの遺伝子座は連続していなければならない事が挙げられる. また, 移民先 S'_k については, S'_i , 及び S'_j における個体の遺伝子長をそれぞれ $n_{i,l}$, 及び $n_{j,l}$ とすると, S'_k における個体の遺伝子長は $(n_i + n_j)l$ である事も必要である.

以上が 4 つの重複方法の詳細である. また, アルゴリズム全体では, 各局所集団で, 交叉・突然変異・選択を行い, 集団間では, 重複・移民の順で処理を行う.

4.3 実験と結果

実験は, 第 2 章で用いた 9 つのテスト関数の 5 次元と 10 次元に対して最小値探索を行う. また, ベースとなるアルゴリズムには SSGA, および PfGA を用い, 遺伝子を連結して他の局所集団へ移民するタイミングは, 1 世代毎に行う. 1 回の試行における終了条件は, 局所集団当たり 10,000 回 (10 次元 Sh と La に対しては 100,000 回) の評価とする.

実験は, 各関数について独立に 300 回の試行を行い, 成功率 SR, 平均評価回数 ENES を求めた. これらの統計量は前述の通り, アルゴリズムの性能を統計的に測る尺度である. これらの量を, PfGA をベースとした遺伝子重複 GA では, 異なる遺伝子重複 (移民) レート (0 ~ 1.0) に対して, SSGA をベースとした遺伝子重複 GA では, 異なる移民個体数 (0 ~ 50) に対して求めた. また, 関数の次元を変化させた場合についても実験を行った. この時の移民レートは, PfGA ベースでは 1, SSGA ベースでは移民個体数 10 である. ただし, この場合の ENES は成功率が 100% のものについてのみ評価を行った.

実験の結果, 重複の効果が全てのタイプで現れたもの, ある特定のタイプのみで効果が現れたもの, 移民による集団の多様性の効果が現れたものに分かれた. 以下, 典型的な結果についてのみ図示し, 説明する.

4.3.1 Sphere model に対する結果

図 4.6, 4.7 は, それぞれ PfGA ベース, および SSGA ベースの GA による 5 次元 Sp に対する ENES である. PfGA ベースの場合は, 全てのタイプで遺伝子重複の効果が顕著に現われ, 重複がない場合 $R = 0$ に比べ, ENES が 1 桁程度も小さくなっている. また, 戦略的にはタイプ B が最も優れており, このときの移民レートは $R = 1$ であることが分かる. SSGA ベースの場合は, 移民数が小さい程 ENES が小さくなっている. 移民数 0 での成功率が 0 であったため, ENES は表示していない. よって図は, 移民数 10 でタイプ B が最も収束性能が良いということを示している. また, 成功率の図は省略するが, 移民がある場合, 全ての

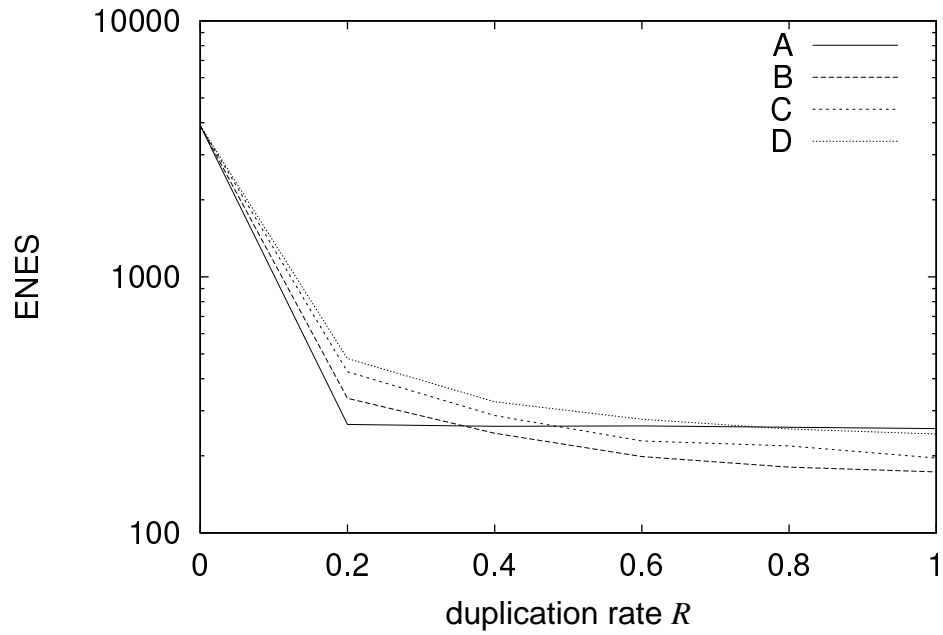


図 4.6: 5次元 S_p における移民レートに対する ENES (PiGA ベース)

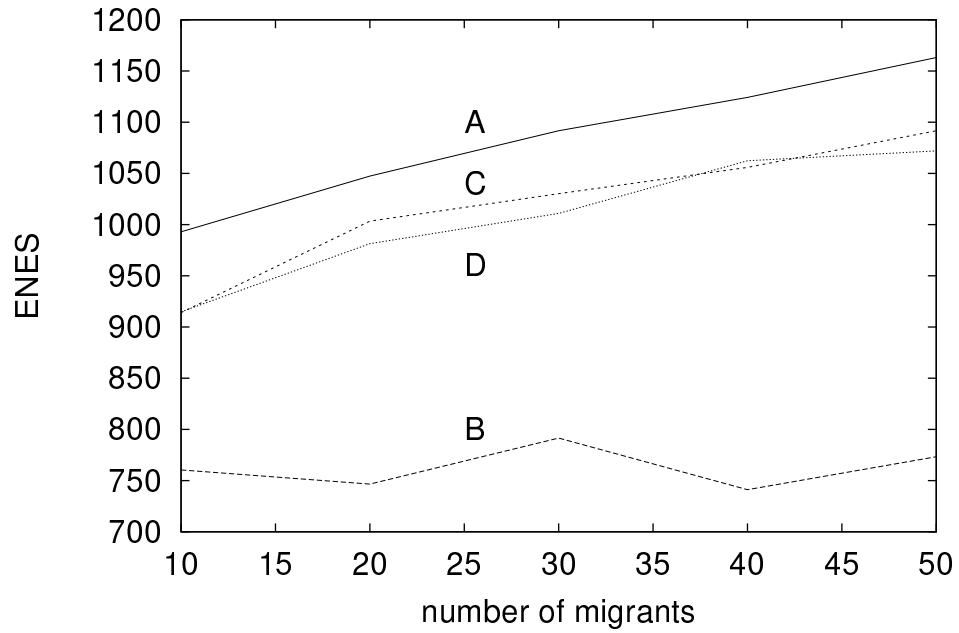


図 4.7: 5次元 S_p における移民数に対する ENES (SSGA ベース)

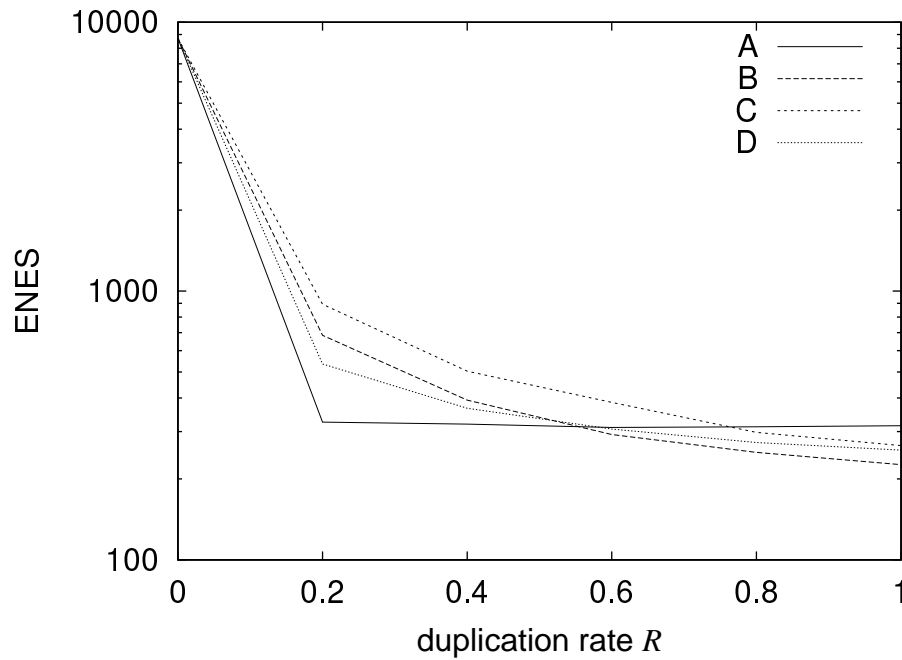


図 4.8: 10次元 S_p における移民レートに対する ENES(PfGA ベース)

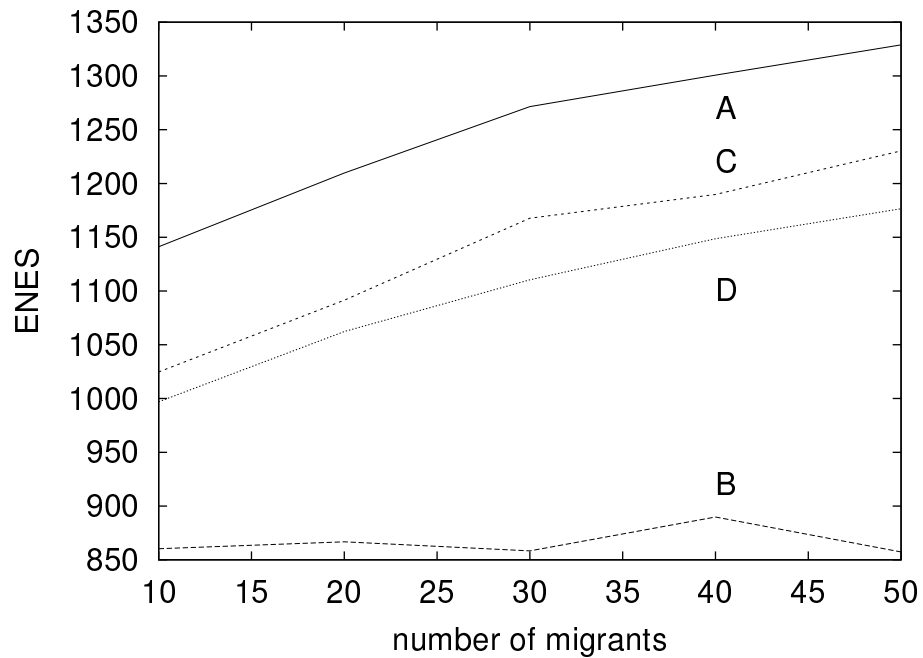


図 4.9: 10次元 S_p における移民数に対する ENES(SSGA ベース)

タイプで 100%となった。10次元の場合も、定性的にはほぼ同様の結果となった。

D_s , R_a に対しての結果も、定性的には S_p に対する結果と同様の結果となった。これらの関数に共通する特徴は対称性があり、変数分離可能という点である。

4.3.2 Randomized Sphere model に対する結果

図 4.10, 4.11 はそれぞれ 5, および 10 次元 R_s に対しての, PfGA ベースの場合の ENES である。このときの成功率はいずれの場合も 100%であったので, ENES から読み取ると, 図より, タイプ A, D で重複の効果が著しく現れたが, B, C ではほとんど効果はなかったことが分かる。また, タイプ D が最も優れた戦略であり, このときの移民レートは $R = 1$ である。

SSGA ベースの場合の ENES についても, タイプ A, D で効果が現われ, 成功率は 100%となったが, タイプ B, C では成功率は 0%という極端な結果となった。

また, 図 4.12, 4.13 はそれぞれ PfGA ベース, SSGA ベースの場合の, R_s に対する ENES の次元依存性を調べたものである。ただし, 関数 R_s の真の最小値は 0 であるので, 簡単のため, 各次元での VTR を全て同じ (10^{-6}) とした。また, PfGA ベースの場合は移民レートは 1, SSGA ベースの場合は移民数を 10 とした。結果は両者共, タイプ A, D で重複の効果が現れ, 次元が大きいところで ENES が次元によらずほぼ一定となった。すなわち, 部分問題に分割して解いているため, 次元には依存しないという結果である。

4.3.3 Michalewicz' function に対する結果

図 4.14, 4.15 はそれぞれ, 5次元 M_i に対する, PfGA ベースの場合の成功率と ENES である。この場合もタイプ A, D で重複の効果が顕著に現われ, 成功率は 100%となった。タイプ C では, 成功率は向上したものの, ENES はタイプ B, C 共にあまり効果は現われなかった。同様に, 図 4.16, 4.17 はそれぞれ, 10次元 M_i に対する成功率, ENES であるが, この場合もタイプ A, D でのみ効果が見られた。

また, SSGA ベースの場合もタイプ A, D でのみ効果が見られた。 R_s , および M_i の 2 つの関数の共通点は, 対称性がなく, 変数分離可能である。タイプ A, D で重複の効果が現れたのは, これらのタイプの部分遺伝子が, 各次元毎の遺伝子に正確に対応しているからである。

4.3.4 Randomized Double Sum に対する結果

図 4.18, 4.19 はそれぞれ, 5次元 R_d に対する, PfGA ベースの場合の成功率と ENES である。この関数に対しては, タイプ D で重複の効果が顕著に現れ, タイプ C ではわずかに現われた。このように成功率が極大値をとるのがこの場合の特徴であり, 最良の移民レート (ここでは 0.4) が存在することを示している。

10次元 R_d に対しては図は省略するが, 5次元の場合と同様にタイプ D で成功率が増加し, 移民レート 0.2 で最大値をとった。また, SSGA ではあまり効果は得られなかった。この関数は, 単峰性, 非対称, 変数分離不可能であることから, 非対称, 変数分離不可能な関数に対しては, タイプ D を用いるという戦略が妥当であると考えられる。

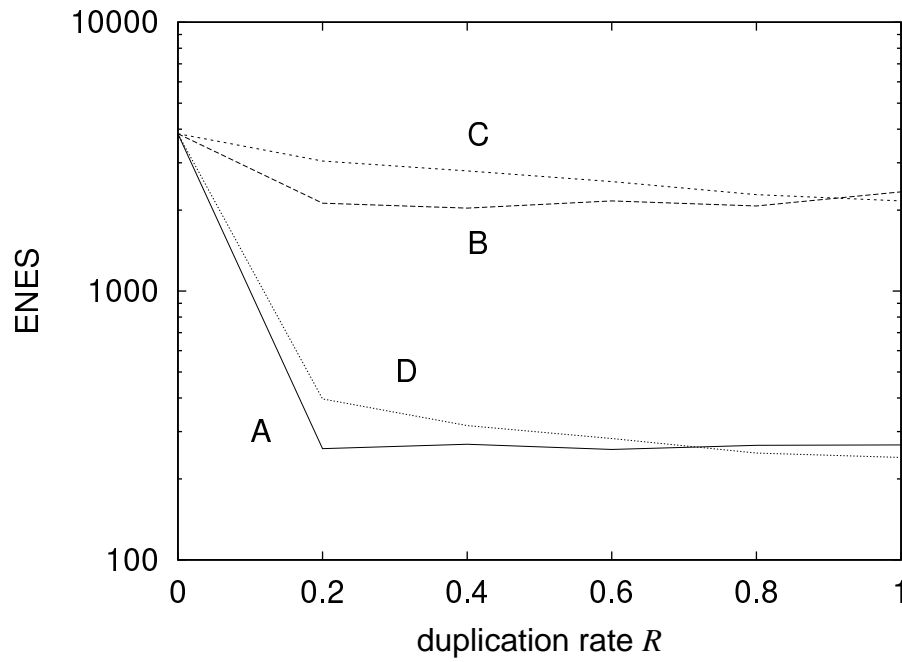


図 4.10: 5次元 R_s における移民レートに対する ENES(PfGA ベース)

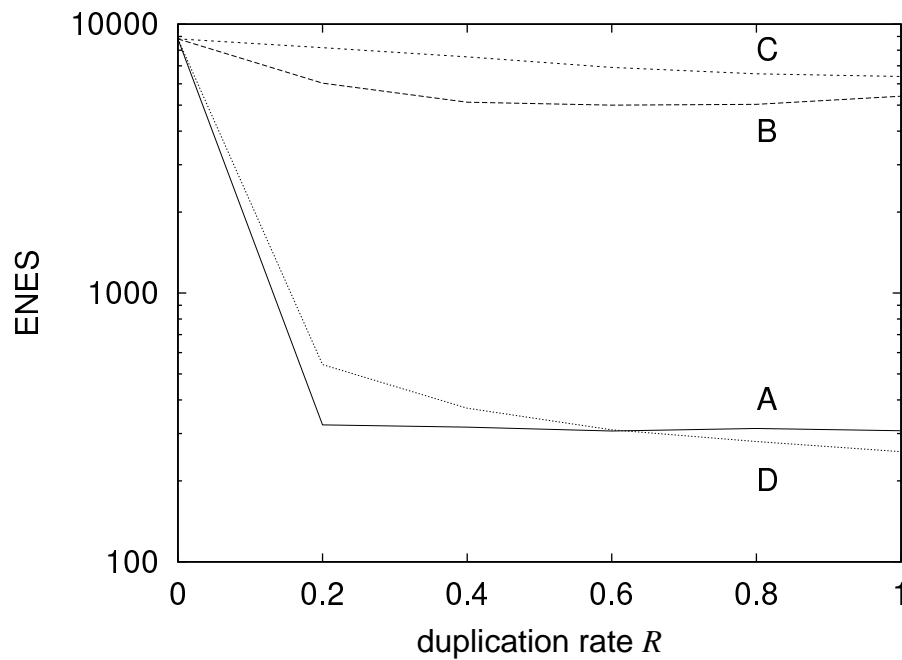


図 4.11: 10次元 R_s における移民レートに対する ENES(PfGA ベース)

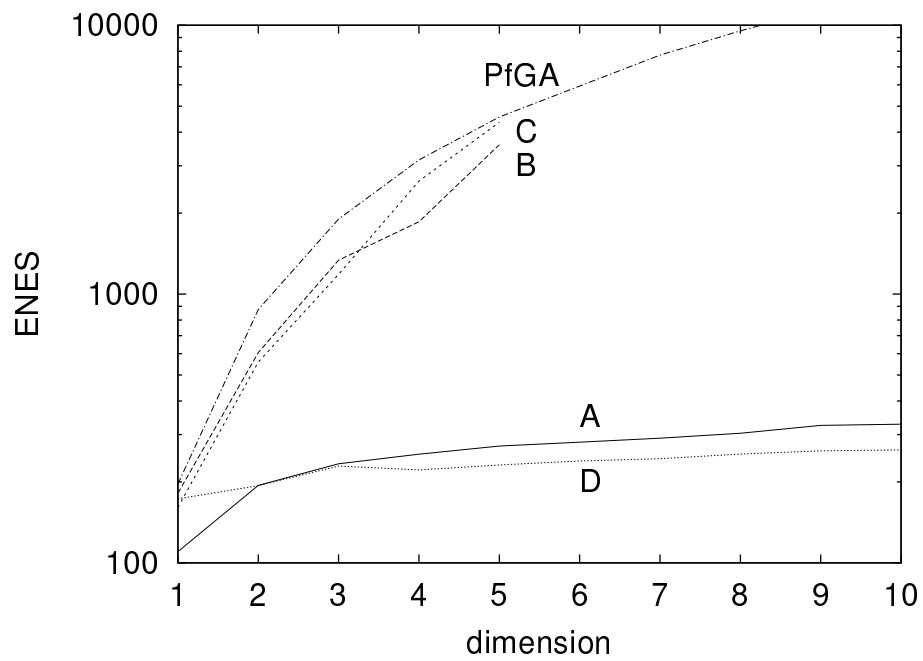


図 4.12: R_s に対する ENES の次元依存性 (PfGA ベース)

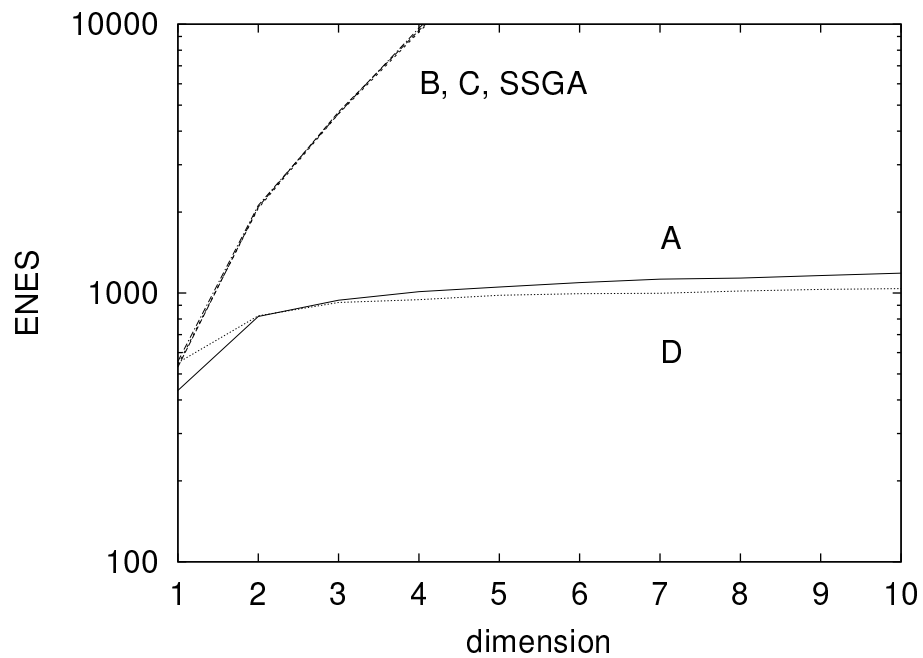


図 4.13: R_s に対する ENES の次元依存性 (SSGA ベース)

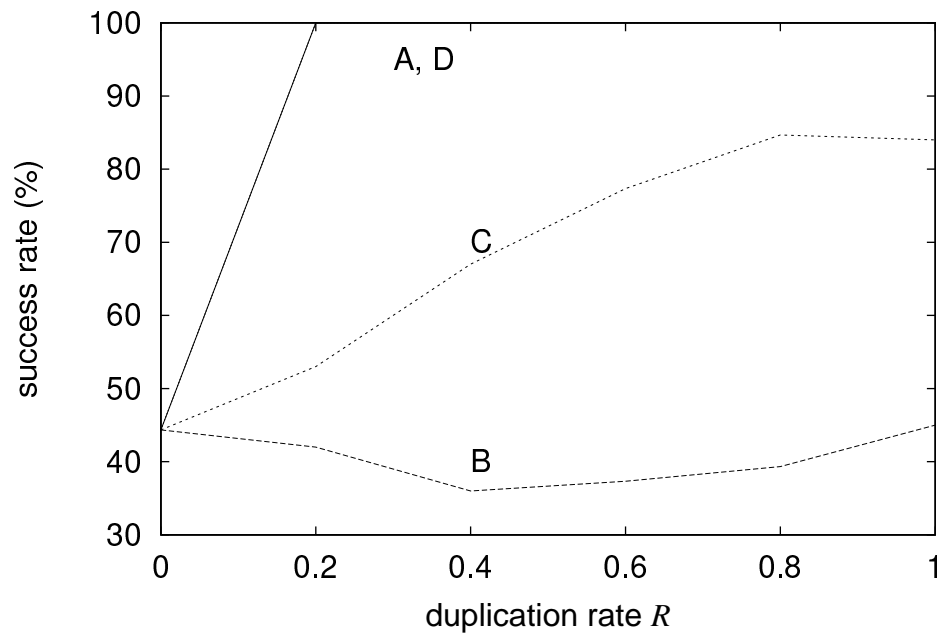


図 4.14: 5次元 Mi における移民レートに対する成功率 (PfGA ベース)

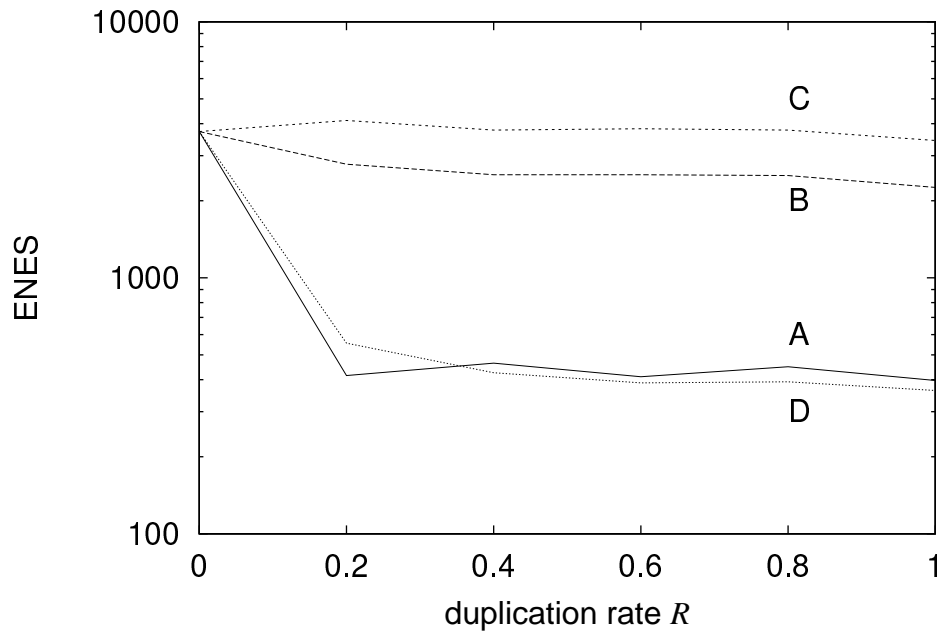


図 4.15: 5次元 Mi における移民レートに対する ENES (PfGA ベース)

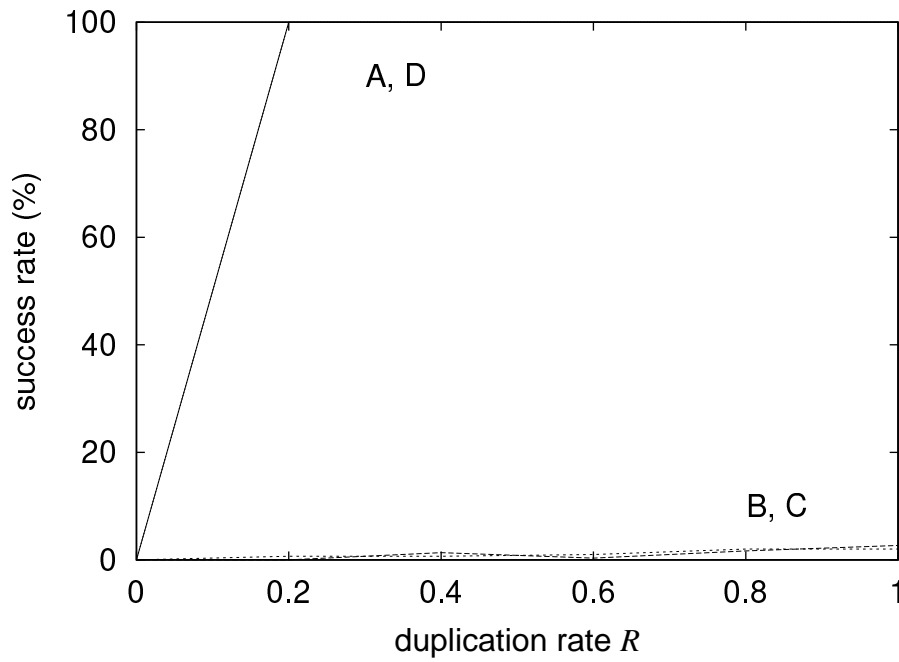


図 4.16: 10次元 Mi における移民レートに対する成功率 (PfGA ベース)

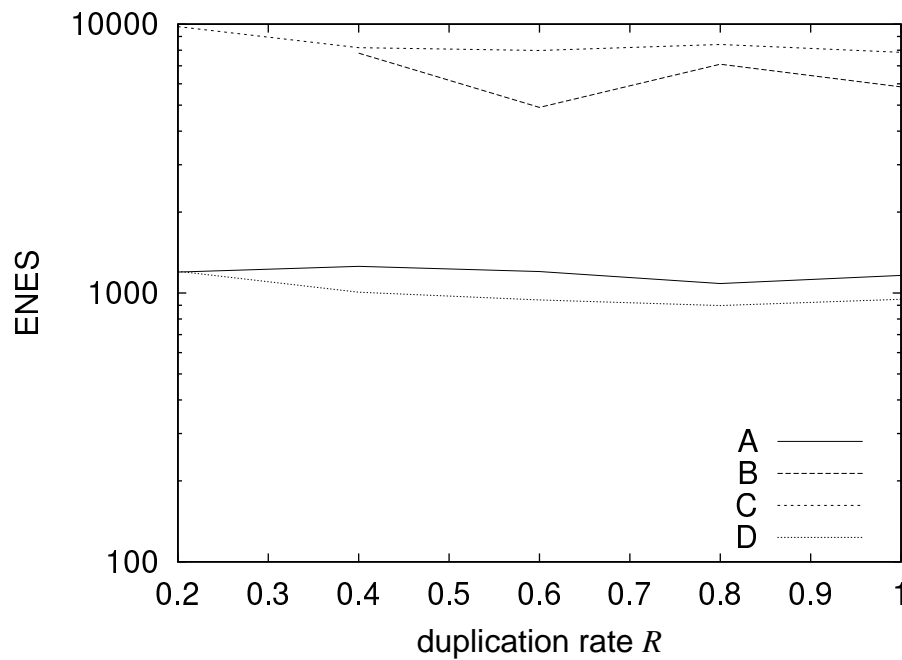


図 4.17: 10次元 Mi における移民レートに対する ENES (PfGA ベース)

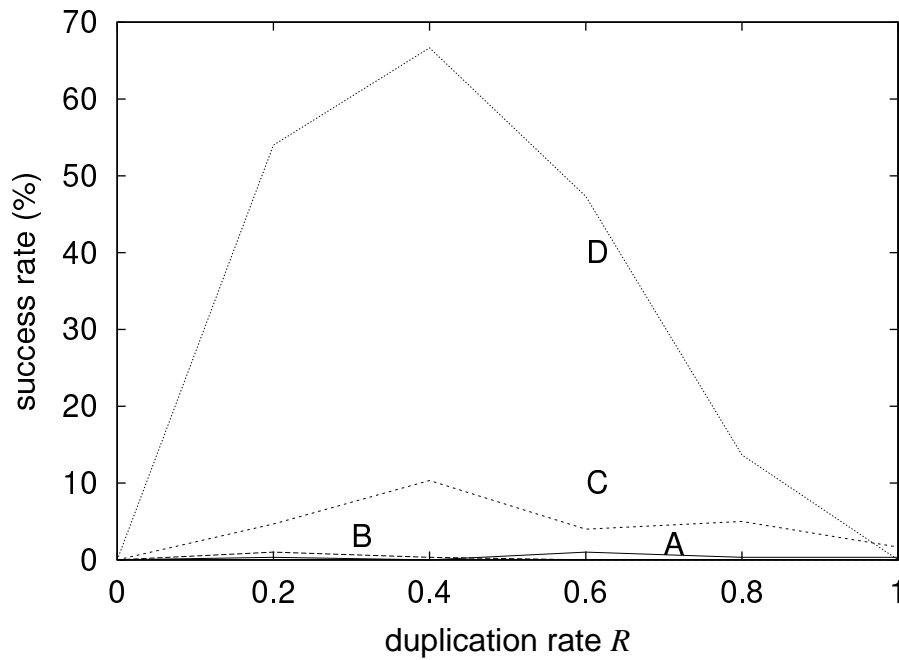


図 4.18: 5次元 Rd における移民レートに対する成功率 (PfGA ベース)

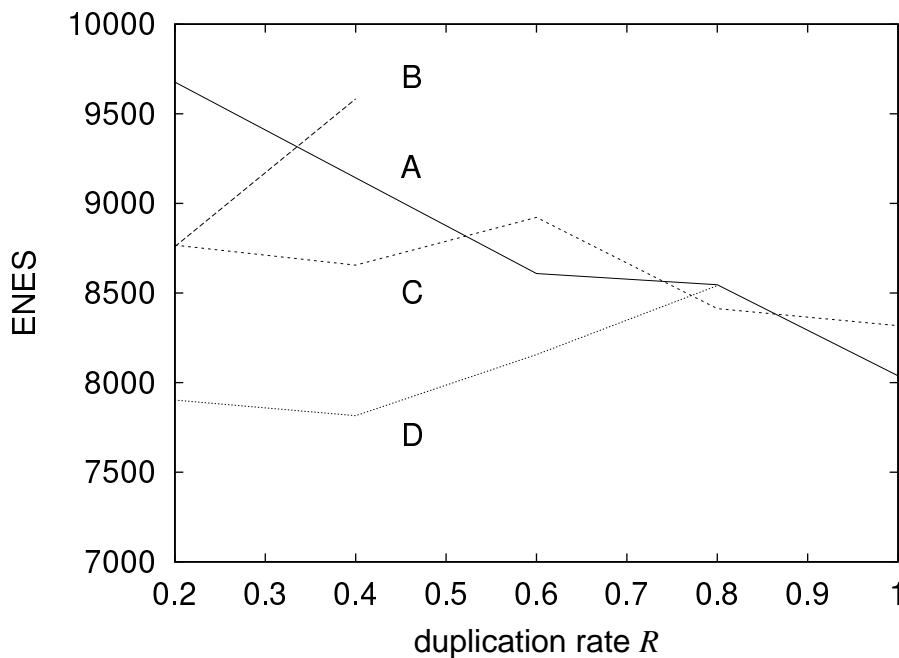


図 4.19: 5次元 Rd における移民レートに対する ENES (PfGA ベース)

4.3.5 Griewank's function に対する結果

図 4.20, 4.21 はそれぞれ, PfGA ベース, SSGA ベースの場合の, 5 次元 Gr に対する成功率である. この関数に対しては, PfGA ベース, SSGA ベース共にタイプ B, C で重複の効果が現れた. 特に SSGA ベースでは, タイプ B で移民個体数 30 のとき成功率が 98% にも達した. ENES については図は省略するが, タイプ B で重複の効果が現われた.

この関数は, 多峰性, 対称, 変数分離不可能であるが, 多峰性関数に対しては SSGA ベースの方が成功率が高いようである. また, 対称性を持つ関数に対しては, タイプ B や C でも, これらの結果のように十分効果が得られることが分かった.

4.3.6 Langerman's function に対する結果

図 4.22, 4.23 は SSGA ベースの場合の, 5 次元 La に対する成功率と ENES である. PfGA ベースの場合には, 重複による移民の効果はほとんど現われなかったが, SSGA ベースでは, もともと高かった成功率はそのまま, 移民数の増加とともに ENES が減少していることが分かる. 戦略的には, タイプ B, C が優れているようである.

Sh に対しても同様の傾向が見られた. これらの関数は多峰性, 非対称, 変数分離不可である. これらの関数に対しては, SSGA をベースとして用い, 移民によって多様性を増大させた結果, ENES を減少させることができたと考えられる.

4.4 実験結果に対する考察と最良戦略

PfGA をベースとした場合 (GD-PfGA) と, SSGA をベースとした場合 (GD-SSGA) との比較を行う. 表 4.1 は全 9 関数の 5, 10 次元に対し, GD-PfGA と GD-SSGA におけるそれぞれの最良のタイプ, そのときの成功率, ENES, および移民レートを列挙したものである. ただしここで言う最良のタイプとは, 成功率が最大値をとる重複方法を指し, 成功率がいずれのタイプでも 100% の場合, あるいはいずれのタイプでもあまり差がない場合は ENES が最も小さいものをとることにする. また右端は, これらの結果から得られる最終的な最良戦略である.

各関数毎に最良のタイプは, GD-PfGA, GD-SSGA 共にほぼ同じである. これは, 遺伝子重複のメカニズムが, GA の個体数に影響されにくい構造を持っているためであり, 逆に目的関数の詳細には非常に敏感であるものと考えられる. また, タイプ A は最良戦略にはならないという点も興味深い. タイプ A は最も簡単な方法であることが原因であると考えられる.

収束性能については, ENES の比較から GD-PfGA の方が全体的に優れている. また, Gr, La に対する成功率の比較から, 多峰性で変数分離不可能な関数に対しては, GD-SSGA の方が向いていると言える. この傾向は重複がないときの PfGA と SSGA との比較結果からも言えることであるが, 重複がない場合とある場合の結果は歴然であり, これは, 重複, 移民による多様性の増大による結果である.

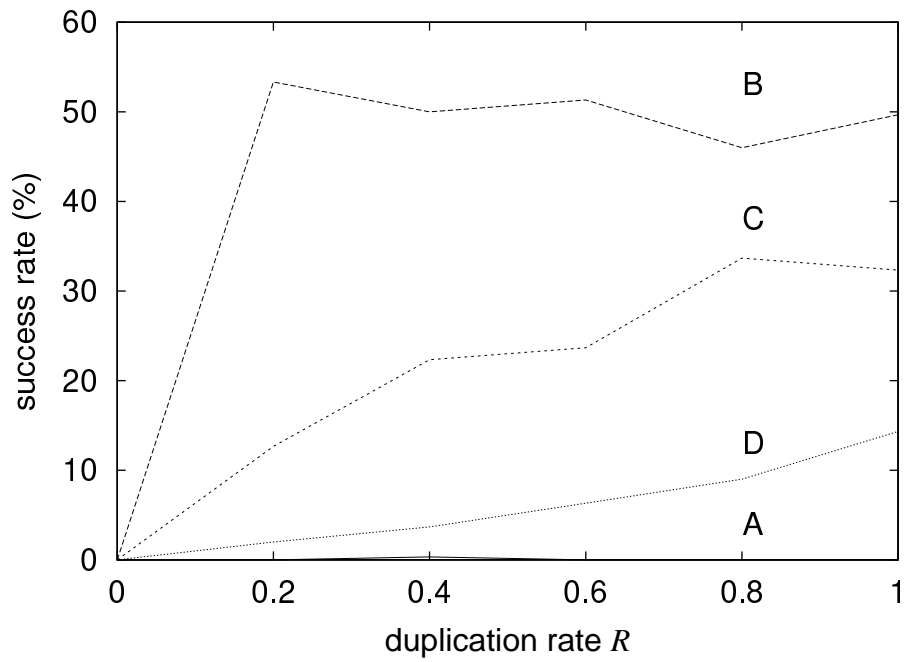


図 4.20: 5次元 Gr における移民レートに対する成功率 (PfGA ベース)

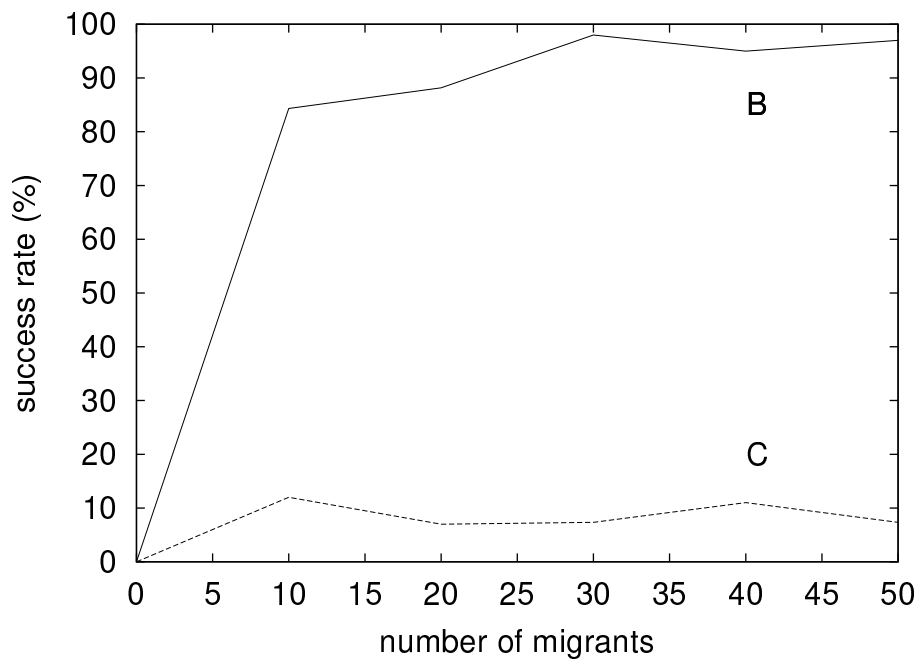


図 4.21: 5次元 Gr における移民数に対する成功率 (SSGA ベース)

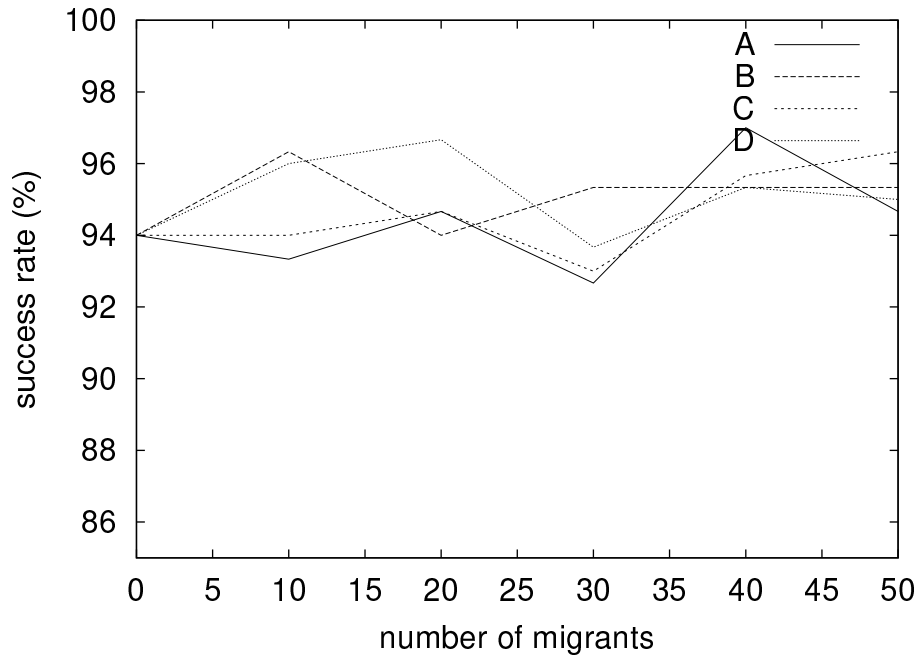


図 4.22: 5次元 La における移民数に対する成功率 (SSGA ベース)

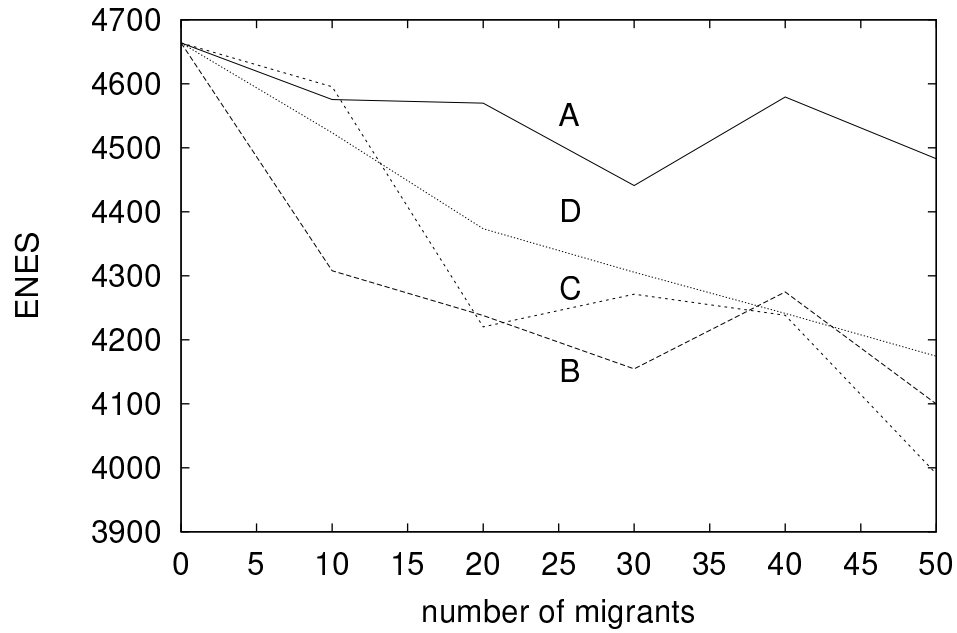


図 4.23: 5次元 La における移民数に対する ENES(SSGA ベース)

表 4.1: PfGA と SSGA をベースとした遺伝子重複 GA の比較結果と最良戦略. (R) は最良戦略における移民レート.

func	feature uni sym sepa	GD-PfGA			GD-SSGA			Best type(base)
		best	succ	ENES(R)	best	succ	ENES(R)	
Sp5	○ ○ ○	B	100%	173 (1)	B	100%	747 (10)	B (PfGA)
Sp10		B	100%	226 (1)	B	100%	857 (10)	
Ds5	○ ○ ×	C	100%	233 (1)	B	100%	1091 (10)	C (PfGA)
Ds10		C	100%	300 (1)	D	100%	1338 (10)	
Rs5	○ × ○	D	100%	227 (1)	D	100%	986 (10)	D (PfGA)
Rs10		D	100%	240 (1)	D	100%	1039 (10)	
Rd5	○ × ×	D	66.7%	7815(0.4)	-	0%	-	D (PfGA)
Rd10		-	0%	-	-	0%	-	
Ra5	× ○ ○	B	100%	196 (1)	B	100%	730 (30)	B (PfGA)
Ra10		B	100%	208 (1)	B	100%	842 (10)	
Gr5	× ○ ×	B	74.3%	1875(0.2)	B	98%	4485 (30)	B (SSGA)
Gr10		D	67.3%	3240(0.4)	D	7.3%	7026 (10)	
Mi5	× × ○	D	100%	362 (1)	D	100%	760 (10)	D (PfGA)
Mi10		D	100%	898 (1)	D	100%	1453 (10)	
Sh5	× × ×	B	9%	4876 (1)	C	1.3%	4606 (10)	C (PfGA)
Sh10		C	2%	5880(0.2)	-	0%	-	
La5	× × ×	C	38%	3859(0.2)	C	97%	3990 (50)	C (SSGA)
La10		C	9%	47579(0.8)	C	54.7%	58248(40)	

4.5 ICEO の他のアルゴリズムとの比較

次に, ICEO の他のアルゴリズムとの比較結果を, 表 4.2 に示す. ただし, 5次元版に対してであり, 参加した8つのアルゴリズムのうち, 上位3位と, PfGA, GD-PfGA, および GD-SSGA の比較である. 各欄は, 上段から ENES, BV(Best Value), RT(Relative Time) の値である (第2章を参照). ICEO では, 20 試行において ENES の小さいほど良いという尺度で比較しているため, GD-PfGA は2位, GD-SSGA は3位に入る. すなわち, 収束性能は PfGA より優れていることが分かる. また BV の値を比較すると, ICEO に参加した上位のアルゴリズムよりも改良されていることが分かる. また, PfGA は RT の値が小さくコンパクトなアルゴリズムであることを示しているが, GDGA の RT は PfGA の RT より大きい. これは, GDGA が PfGA に比べて個体の移民などの計算量があるためである.

表 4.2: 第 1 回 ICEO と PfGA, GD-PfGA, および GD-SSGA の比較

ENES BV RT	Bilchev and Parmee	Li and Smith	Storn and Price	PfGA	GD-PfGA	GD-SSGA
Sphere Model (Sp)	20 3.88e-15 2	243 0.0 12.7	736 - 4.67	4,067 0.0 0.91	173 0.0 2.39	723 0.0 71.26
Griewank's function (Gr)	41 7.99e-6 2	21,141 1.69e-5 3.1	5,765 - 1.79	6,785 4.66e-7 0.90	1367 3.65e-10 2.35	2537 3.65e-10 70.76
Shekel's foxholes (Sh)	74 -10.327 2	6,318 -10.403 0.25	76,210 - 0.80	1,619 -10.40 0.33	456 -10.404 2.40	3632 -10.396 41.47
Michalewicz' function (Mi)	120 -4.6876 2	6,804 -4.687 1.28	1,877 - 1.11	5,131 -4.688 0.90	240 -4.68766 3.54	786 -4.68766 79.24
Langerman's function (La)	176 -1.499 2	4,131 -1.499 1.62	5,308 - 1.35	5,274 -1.499 0.43	2,330 -1.499 2.12	3,460 -1.499 52.83

4.6 まとめ

本章では、大野乾により 1970 年代に提案された遺伝子重複説にヒントを得た進化的計算法 (Gene Duplicated GA: GDGA) を提案した。局所集団を進化させるベースとなる GA としては、パラメータフリー GA (PfGA) と、定常状態 GA (SSGA) の 2 つを比較検討した。また遺伝子の重複方法については、4 つのタイプを提案した。提案した進化的計算法の性能を評価するために、関数最適化ベンチマーク問題を用い、最適解に到達する成功率と収束性能などの比較を行い、遺伝子重複 GA の有効性を検討した。

その結果、関数の特徴である単峰/多峰性、対称/非対称性、変数分離可能/不可能性を組み合わせた関数に対して、遺伝子重複の効果が現れ、成功率を向上させることができた。国際コンテストとの比較においては、解の精度が向上することも確認された。各関数ごとに適切な遺伝子重複のタイプはベースとなる GA に依存せず、また、概してベースとなる GA は SSGA よりも PfGA が優れていること、多峰性、非対称性、変数分離不可能性を同時に持つ関数に対しては、遺伝子重複による成功率の向上は望めないが、SSGA のような多数個体ベースの GA を用い、移民個体数の増加による多様性の増大によって、収束性能の向上が期待できることが確認できた。

最後に、実験で得られた情報を経験則として今後も活用できるように、項目別にまとめる。

4.6.1 ベースとなる GA に関して

ベースとなる GA については、関数の極小値の数に応じて個体数を選択するのがよい。すなわち、単峰性関数、あるいは少数の極小値を持つ関数では、収束の速さを重視して少数個体の GA(ここでは PfGA) を、また、多峰性関数では成功率を上げるために多数個体の GA(ここでは SSGA) を用いるのがより良い戦略であるといえる。

4.6.2 遺伝子重複のタイプに関して

最良の重複方法は、ベースとなる GA にはあまり依存しない。個々の重複方法については、対称性のある関数では基本的にどのタイプでも重複の効果が現れ、対称性のない関数であっても変数分離可能ならば、タイプ A, D で重複の効果が現れる。また、非対称、変数分離不可能な関数に対しては、単峰性ならばタイプ D、多峰性ならばタイプ C を用いるのがより良い戦略である。

4.6.3 移民レートに関して

単峰性関数に対しては、PfGA ベースの GDGA では移民レートはなるべく大きい方が ENES が小さく、収束性能は良い。逆に SSGA ベースの GDGA では移民個体数は小さい方が良い。多峰性関数に対しては、その関数自体によっても異なるが、ベースとなる GA にあまり依存せず、移民レートは一般に大きい方が性能が良い。この理由は、多峰性関数に対しては移民個体数が多い程、系全体の多様性が増すためである。

4.6.4 次元依存性に関して

変数分離可能な関数に対しては、次元の増加に対して ENES はほとんど増加することなく一定である。これは、問題を部分問題に分割して解くことができるためである。

4.6.5 並列分散処理との比較

第3章で述べた並列分散処理との比較に関しては、単峰性関数に対しては PfGA ベースの GDGA の方が、並列 PfGA(UD1) よりも収束が速く、成功率も高い。しかしながら多峰性関数に対しては逆であり、並列 PfGA の方が収束が速い事が分かる。一般的に、多峰性関数に対する探索は、個体の多様性が重要である。このことから、並列 GA における個体の多様性は、遺伝子重複における個体の多様性よりも大きいことが言える。

第5章 モンテカルロ法による確率探索と 遺伝的アルゴリズムとの比較

5.1 概要

本章では、統計力学的計算法と進化的計算法との関係を明らかにする目的で、モンテカルロ法 (MC: Monte Carlo Method) と遺伝的アルゴリズム (GA) との比較を行う。MC は個体数が1個の GA に相当していると考えられる。すなわち、突然変異のみで状態更新していくアルゴリズムであるため、GA の1種であるとみなしても差し支えない。そこで、両者を比較するために、第2章で定義した関数最適化問題を適用し、MC の性能評価を行い、GA と比較する。これによって、確率的探索アルゴリズム全般の総合的な評価が可能となることが期待できる。

MC については、その拡張であるシミュレーテッドアニーリング (SA: Simulated Annealing)、およびメトロポリス法 (MP: Metropolis Method) を用い、また、MC の別の拡張方法である量子モンテカルロ法 (QMC: Quantum Monte Carlo) の計算方法について提案する。従来法における目的関数はスピン系の自由エネルギーであるが、SA を用いることにより、エネルギー関数の最小値が求まることが保証されている。また、他の問題にも適用できるように拡張したものが MP である。これは、ある状態における関数値と、摂動を加えた後の関数値の差のみを用いるため、どのような問題にも非常に簡単に適用できるという利点がある。また、QMC を導入する理由は、MC で用いられる温度揺らぎを量子揺らぎに置き換えることによってどのような効果が得られるかを明確にし、これにより探索効率を向上させるためである。

5.2 モンテカルロ法のエネルギー関数最小化問題への適用に関する従来手法

5.2.1 イジングモデルのエネルギー最小化問題

ここでは、MC の起源となる、スピン系の熱力学特性について述べる。系の内部に空間的に固定されていて、 $|\uparrow\rangle, |\downarrow\rangle$ の2状態、あるいは ± 1 の2値をとるような要素をスピン (spin) と呼ぶ。このようなスピンの構成される系はスピン系と呼ばれ、磁性体のモデルとしてよく用いられる。各スピンは近傍のスピンと交換相互作用 (exchange interaction) によって作用し合い、自身の状態を変更させる。最も簡単なモデルはイジングモデル (Ising model) と呼ばれ、例えば、強磁性イジングモデルでは交換相互作用は正であり、基底状態は、スピンの

揃っている状態である。逆に反磁性イジングモデルでは交換相互作用は負であり、スピンの上下が交互に並んでいる状態が最もエネルギーが小さくなる。

イジングモデルのエネルギー関数(ハミルトニアン: Hamiltonian) は次式で定義される。

$$H = - \sum_{i \neq j} J_{ij} S_i S_j \quad (5.1)$$

ただし, J_{ij} はスピン S_i と S_j の間に働く交換相互作用であり, $J_{ij} > 0$ で強磁性体, $J_{ij} < 0$ で反磁性体を表す. i, j のとり方は, 磁性体では最近接格子の組であるが, ニューラルネットワークなどでは, とり得る全ての組を表す場合もある。

熱統計力学によると, ある系が外部と粒子の交換がなく, エネルギーのみ交換するような場合, この系はカノニカル集団 (canonical ensemble) と呼ばれ, 熱平衡状態では以下の Gibbs 分布 (カノニカル分布) になる。

$$P_a = \frac{1}{Z} \exp(-\beta E_a) \quad (5.2)$$

$$Z = \sum_a \exp(-\beta E_a) \quad (5.3)$$

ただし, P_a は系が状態 a にある確率で, その状態のエネルギーを E_a とする. β は温度の逆数であり, $\beta = 1/T$ である. また, Z は分配関数 (partition function) と呼ばれ, 熱平衡状態における系の巨視的な量はほぼ全て, この分配関数より導かれる. しかしながらこの分配関数の計算は非常に困難であり, 厳密解が得られるのはごく限られたモデルに対してだけである。

式 (5.2) に対し, 量子統計力学では例えば P_{ab} の様に非対角成分が 0 でない場合を取り扱うことが多い. そこで次式の様に分布関数を行列を用いて表現する。

$$\rho = \frac{1}{Z} \exp(-\beta H) \quad (5.4)$$

$$Z = \text{Tr} \exp(-\beta H) \quad (5.5)$$

ここで H は系のハミルトニアンである. これを用いると, 状態 a から b への遷移確率 P_{ab} や, ある物理量 A の期待値 $\langle A \rangle$ などは,

$$\begin{aligned} P_{ab} &= \langle a | \rho | b \rangle \\ &= \frac{1}{Z} \langle a | \exp(-\beta H) | b \rangle \end{aligned} \quad (5.6)$$

$$\begin{aligned} \langle A \rangle &= \text{Tr} \rho A \\ &= \frac{1}{Z} \text{Tr} \exp(-\beta H) A \end{aligned} \quad (5.7)$$

のように計算される. ただし Tr は行列のトレースであり, 対角成分の和をとることを意味する. イジングモデルのハミルトニアン (5.1) は対角的であるので, 式 (5.2) を用いても計算できるが, 本章で扱うモデルに対しても一般性を失わないように, このような表現を使用する。

スピン S_i の期待値は、このハミルトニアンの中で S_i が含まれる部分を H_i とおき、それ以外を H' とすると、 $H = H' + H_i$ であるから、

$$\begin{aligned}
\langle S_i \rangle &= \frac{1}{Z} \text{Tr} \exp(-\beta H) S_i \\
&= \frac{1}{Z} \text{Tr} \exp(-\beta H') \exp(-\beta H_i) S_i \\
&= \frac{1}{Z} \text{Tr}' \exp(-\beta H') \text{Tr}_i \exp(-\beta H_i) S_i \\
&= \frac{1}{Z} \text{Tr}' \exp(-\beta H') \sum_{S_i} \exp(-\beta H_i) S_i \\
&= \frac{1}{Z} \text{Tr}' \exp(-\beta H') [\exp(-\beta H_i(S_i = 1)) - \exp(-\beta H_i(S_i = -1))] \\
&= \frac{1}{Z} \text{Tr}' \exp(-\beta H') \frac{\exp(-\beta H_i(S_i = 1)) + \exp(-\beta H_i(S_i = -1))}{\exp(-\beta H_i(S_i = 1)) + \exp(-\beta H_i(S_i = -1))} \\
&\quad \times [\exp(-\beta H_i(S_i = 1)) - \exp(-\beta H_i(S_i = -1))] \tag{5.8}
\end{aligned}$$

となる。 $H_i(S_i = \pm 1) = \mp \sum_{j \neq i} J_{ij} S_j$ であるから結局、

$$\begin{aligned}
\langle S_i \rangle &= \frac{1}{Z} \text{Tr}' \exp(-\beta H') \sum_{S_i} \exp(-\beta H_i) \tanh \left(\beta \sum_{j \neq i} J_{ij} S_j \right) \\
&= \frac{1}{Z} \text{Tr} \exp(-\beta H) \tanh \left(\beta \sum_{j \neq i} J_{ij} S_j \right) \\
&= \left\langle \tanh \left(\beta \sum_{j \neq i} J_{ij} S_j \right) \right\rangle \\
&\sim \tanh \left(\beta \sum_{j \neq i} J_{ij} \langle S_j \rangle \right) \tag{5.9}
\end{aligned}$$

が得られる。最後の近似は、高次の相関を無視するもので、例えば $\langle S_i S_j \rangle = \langle S_i \rangle \langle S_j \rangle$ などと仮定すると得られる。これは切断近似、あるいは平均場近似 (MFA: mean field approximation) と呼ばれているものであり、強磁性体のような揺らぎの小さな系で良い近似であるとされている。式 (5.9) は、スピン S_i の向きが、 S_i 以外のスピンからの寄与の和によって「確率的に」決定されるということの意味している。

5.2.2 イジングモデルの緩和過程のシミュレーション

ここでは、イジングモデルの緩和過程のシミュレーション方法について述べる。緩和 (relaxation) とは、エネルギーの高い状態から低い状態への遷移を指す。初期状態をランダムなスピン配列とすると、系は一般に熱平衡状態 (ground state) ではなく、励起状態 (excited state) である。

緩和のメカニズムは、次の現象論的なマスター方程式からスピンの運動方程式を導くことにより理解される。

$$\dot{P}_{\{S_i\}}(t) = \sum_i [\Gamma_{-S_i, S_i} P_{-S_i}(t) - \Gamma_{S_i, -S_i} P_{S_i}(t)] \tag{5.10}$$

$$\Gamma_{\downarrow i \uparrow i} = 1 \quad (5.11)$$

$$\Gamma_{\uparrow i \downarrow i} = \exp\left(-\beta \sum_j J_{ij} S_j\right) \quad (5.12)$$

$$\langle \dot{S}_i \rangle = \gamma_i \left[\tanh\left(\beta J \sum_{j \neq i} \langle S_j \rangle\right) - \langle S_i \rangle \right] \quad (5.13)$$

これは最も簡単な緩和方程式であり、緩和時間 $1/\gamma_i$ でスピンは指数関数的に緩和し、熱平衡状態に達するのである。マスター方程式の導出については省略するが、スピン系と熱浴（フォノン系）との相互作用を考慮することにより、導かれる。

このスピンの運動方程式を用いて、実際の物理系を単純化して模倣するのがモンテカルロシミュレーションである。イジングモデルのエネルギーを下げる緩和過程の実現方法は、式(5.9)の右辺を $[0, 1]$ に射影し、乱数と比較することにより、確率的にスピンの向きを決めるという方法である。すなわち、

1. スピンの番号 i を選ぶ。
2. $[0, 1]$ の一様乱数 r を発生し、

$$r < \frac{1}{2} \left[1 + \tanh\left(\beta \sum_{j \neq i} J_{ij} S_j\right) \right] \quad (5.14)$$

ならば $S_i = +1$ 、そうでなければ $S_i = -1$ とする。

である。以上を繰り返すことにより、系は式(5.2)の熱平衡状態に達する。この方法は古くから1次元、2次元、3次元のスピン系において、様々な境界条件や、初期条件の下で実験されてきた。また、スピンの和で表される磁化 (magnetization) や、自由エネルギー (free energy)、エントロピー (entropy)、比熱 (specific heat) などのマクロな物理量の計算や、相転移 (phase transition) なども研究されてきた。統計力学以外では、ニューラルネットワークの分野でパターンの記憶と想起などが研究されてきた。

5.2.3 シミュレーテッドアニーリングによる最適化

MCを用いて系の時間発展をシミュレートすると、熱平衡状態においては、Gibbsの自由エネルギー

$$F = \langle H \rangle - TS \quad (5.15)$$

が最小となっている。ただし S はエントロピーである。第1項が最小化するべき目的関数であるため、物質を徐冷して結晶を作る方法にヒントを得て、ゆっくりと $T \rightarrow 0$ に近付ける方法が考えられてきた。これが、シミュレーテッドアニーリング (SA: Simulated Annealing) である。この方法によって緩和させると、十分時間が経過した後の状態が基底状態となり、エネルギー関数(5.1)が最小となる。

計算例として、図5.1に1次元イジングモデルに対する緩和過程のシミュレーション結果を示す。横軸が更新回数、縦軸が基底エネルギーとの差であり、 $E_0 = -(N-1)J$ である。こ

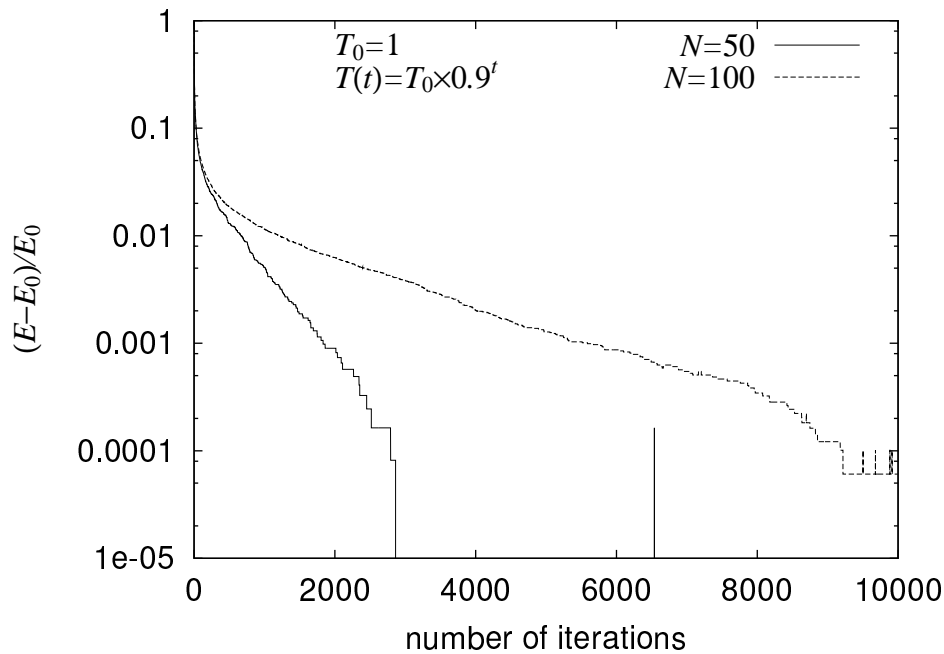


図 5.1: 1次元強磁性イジングモデルに対するエネルギーの緩和過程. 縦軸は基底エネルギーとの差, 横軸は更新回数, サイズは $N = 50$, および $N = 100$, 初期温度は $T_0 = 1$, 温度スケジュールは $T(t) = T_0 \times 0.9^t$.

ここで, $J_{ij} = J = 1$ としている. これは強磁性体である. 1回の更新において, N 個の乱数を発生させ, 任意のスピンを反転するかどうかの評価を行うという方法を採用した. サイズは $N = 50$, および $N = 100$, 温度スケジュールは $T(t) = T_0 \times 0.9^t$ であり, $T_0 = 1$ である. また, サンプル数はサイズの10倍とし, 独立に実行したものを更新回数毎に平均した. 問題となる境界条件については, 開放端, すなわち $S_0 = 0$, および $S_{N+1} = 0$ とした. 図から, エネルギーの緩和曲線は, サイズが小さい程緩和が速いことが分かる. またこの曲線は, t の大きいところで指数的な緩和であるとみなせる. これを $e^{-t/\tau}$ であると仮定し, 具体的な緩和時間 τ を見積もると, $N = 50$ の時, $\tau \sim 10^3$, $N = 100$ の時, $\tau \sim 2 \times 10^3$ となる.

本節では, 目的関数をイジングモデルのエネルギーとし, これを最小化するための従来手法として, MCとSAを併用する方法を用いた. これらはシミュレーションの方法であると同時に探索アルゴリズムそのものである. よって緩和時間の見積りに関しては, これが直接収束性能を表す指標となることを考慮して記述したものである.

5.3 量子モンテカルロ法による最適化

MCやSAの探索性能を向上させるためには, 初期温度, およびアニーリングスケジュールの調節が必要であるが, その他の方法としては, 温度揺らぎを用いた確率的探索を, 量子揺らぎを用いた確率的探索に変更するものがある. これを量子モンテカルロ法(QMC: Quantum Monte Carlo Method)と呼ぶ. QMC自体は, 量子スピン系のエネルギーを最小化する目的

で導入されたという歴史があるが、量子揺らぎを徐々に小さくしていく方法を採用することにより、イジングモデルのエネルギー最小化問題にも適用できると考えられる。

本節では、QMCの従来法である、量子スピン系を古典的スピン系に置き換えてMCを適用する方法と、著者が提案する、平均場近似(MFA)を用いたQMCについて説明し、実際にシミュレーションを行って、この方法の有効性を示す。目的はイジングモデルのエネルギーの最小値を求めるための方法として、温度や量子効果がどのように関与してくるかを調べることにある。

5.3.1 量子スピン系を古典的スピン系に置き換える従来法

量子スピン系とは、量子力学的なスピン演算子を含むスピン系のことであり、TIモデルやXYモデルと呼ばれるモデルが古くから研究されてきた。また、実際の物理系においても、量子揺らぎを無視することのできない大きさの素子から構成されるデバイス [34] などが、TIモデルで記述されることを見出した [35]。このように、量子スピン系は身近な存在となりつつある。

TIモデル [36][37][38] は以下の通りである。

$$H = H_1 + H_2 \quad (5.16)$$

$$H_1 = -J \sum_{i=1}^{N-1} S_i^z S_{i+1}^z \quad (5.17)$$

$$H_2 = -H_x \sum_{i=1}^N S_i^x \quad (5.18)$$

H_1 がイジングモデルのエネルギー、 H_2 が量子揺らぎに相当する項であり、 S_i^x は i 番目のスピンを反転させる演算子である。すなわち、

$$S_i^x |S_i\rangle = | - S_i\rangle \quad (5.19)$$

である。このモデルは、 x 方向の磁場中にあるイジングモデルという意味から、TIモデル (Transverse Ising model) と呼ばれている。1次元TIモデルの分配関数は、指数部分を小さく分割すると、近似的に交換関係は0であるという Trotter の公式 (Trotter's formula) を用いて、

$$\begin{aligned} Z &= \text{Tr} \exp[-\beta (H_1 + H_2)] \\ &= \lim_{M \rightarrow \infty} \text{Tr} [\exp(-\beta H_1/M) \exp(-\beta H_2/M)]^M \end{aligned} \quad (5.20)$$

と書ける。この式に完全系

$$\sum_{\{S_i\}} |\{S_i\}\rangle \langle \{S_i\}| = 1 \quad (5.21)$$

を $M-1$ 個挿入し、また、 S^x に対する次の関係式

$$\langle S | e^{aS^x} | S' \rangle = \left\langle S \left| \sum_{n:\text{even}} \frac{a^n}{n!} + \sum_{n:\text{odd}} \frac{a^n}{n!} S^x \right| S' \right\rangle$$

$$\begin{aligned}
&= \frac{1 + SS'}{2} \cosh a + \frac{1 - SS'}{2} \sinh a \\
&= \frac{1}{2} (e^a + SS' e^{-a}) \\
&= \frac{1}{2} \sqrt{\frac{e^{2a} - e^{-2a}}{(e^a + e^{-a})(e^a - e^{-a})}} (e^a + SS' e^{-a}) \\
&= \sqrt{\frac{1}{2} \sinh 2a (\coth a)^{SS'}} \\
&= \left(\frac{1}{2} \sinh 2a\right)^{1/2} \exp\left(\frac{1}{2} SS' \ln \coth a\right) \tag{5.22}
\end{aligned}$$

を用いて,

$$\begin{aligned}
Z &= \lim_{M \rightarrow \infty} \sum_{\{S_{i,j}\}} \langle \{S_{1,j}\} | \exp(-\beta H_1/M) \exp(-\beta H_2/M) | \{S_{2,j}\} \rangle \\
&\quad \langle \{S_{2,j}\} | \exp(-\beta H_1/M) \exp(-\beta H_2/M) | \{S_{3,j}\} \rangle \\
&\quad \vdots \\
&\quad \langle \{S_{M,j}\} | \exp(-\beta H_1/M) \exp(-\beta H_2/M) | \{S_{1,j}\} \rangle \\
&= \lim_{M \rightarrow \infty} \sum_{\{S_{i,j}\}} \exp \left[\frac{\beta J}{M} \sum_{i,j} S_{i,j} S_{i,j+1} \right] \\
&\quad \times \left[\frac{1}{2} \sinh \left(\frac{2\beta H_x}{M} \right) \right]^{1/2} \exp \left[\frac{1}{2} S_{i,j} S_{i+1,j} \ln \coth \left(\frac{\beta H_x}{M} \right) \right] \\
&= \lim_{M \rightarrow \infty} C_M \text{Tr} \exp \left[\frac{\beta J}{M} \sum_{i=1}^M \sum_{j=1}^{N-1} S_{i,j} S_{i,j+1} + K_M \sum_{i=1}^M \sum_{j=1}^N S_{i,j} S_{i+1,j} \right] \tag{5.23}
\end{aligned}$$

$$C_M = \left[\frac{1}{2} \sinh \left(\frac{2\beta H_x}{M} \right) \right]^{NM/2} \tag{5.24}$$

$$K_M = \frac{1}{2} \ln \coth \left(\frac{\beta H_x}{M} \right) \tag{5.25}$$

と書き直される [39][40]. 1次元 TI モデルは, 新たに加わった $M - 1$ 個の完全系が, 等価的に新たなスピン方向 (Trotter direction) を表している. すなわち, 1次元 TI モデルは, 分配関数の計算においては, 2次元の古典的イジングモデルとして表現されることが分かる. 同様の計算を行うと, d 次元 TI モデルは $d + 1$ 次元の古典的イジングモデルと等価となる. また, 新たなスピン方向の境界条件は周期的である. この式中で, 第1項の係数は M の増加に従って減少し, 第2項の係数は M の増加に従って増加する. これは, 第2項が量子遷移を表す項であり, 本来の1次元方向のスピンの相互作用よりも新たな方向のスピンの相互作用の方が寄与が大きくなることを意味する. これを用いてスピン S_i の期待値を求めると,

$$\begin{aligned}
\langle S_i \rangle &= \frac{1}{Z} \sum_{\{S_{1,i}\}} \langle \{S_{1,i}\} | S_i^z [\exp(-\beta H_1/M) \exp(-\beta H_2/M)]^M | \{S_{1,i}\} \rangle \\
&= \lim_{M \rightarrow \infty} \left\langle \tanh \left[\frac{\beta J}{M} (S_{1,i-1} + S_{1,i+1}) + K_M (S_{M,i} + S_{2,i}) \right] \right\rangle
\end{aligned}$$

$$\sim \lim_{M \rightarrow \infty} \tanh \left[\frac{\beta J}{M} (\langle S_{1,i-1} \rangle + \langle S_{1,i+1} \rangle) + K_M (\langle S_{M,i} \rangle + \langle S_{2,i} \rangle) \right] \quad (5.26)$$

となる。前述のイジングモデルにおけるスピンの期待値と比較して、第2項に新たに加わった部分が量子遷移の項である。これを、 M を十分大きくとり、2次元イジングモデルとしてMCを適用する方法がQMCの従来法である。しかしながらこの計算では、精度を上げるために M を大きくしなければならず、また、系が緩和して熱平衡状態に達するための計算量は単に M 倍ではなく、それ以上であることが、2次元イジングモデルの詳細なサイズ効果の研究から知られている。すなわち、計算量が増えることがこの計算法の欠点である。

5.3.2 平均場近似を用いた量子モンテカルロ法

従来法とは異なる方法で、平均場近似 (MFA: Mean Field Approximation) を用いた計算方法を提案する。これはイジングモデルの箇所でも触れたが、スピンの高次の相関を無視する近似である。言い換えると、1個のスピンの遷移規則は、全系のスピンのみに依存するものである (コヒーレンスがある) が、これを近傍のスピンのみに依存するように近似するのである。

まずハミルトニアン of 交換相互作用項を次のように書き換える。

$$H_1 = -\frac{J}{2} \sum_{i=1}^N S_i^z \langle S_{i-1} + S_{i+1} \rangle \quad (5.27)$$

ただし、 S_0, S_{N+1} は境界条件を表すものとする。この近似は注目するスピンの近傍のスピンを、その期待値で置き換えるものである。そうすると、このハミルトニアンの厳密対角化により、スピンの期待値が次のように求まる。

$$\langle S_i \rangle = \frac{J \langle S_{i-1} + S_{i+1} \rangle / 2}{E_i} \tanh(\beta E_i) \quad (5.28)$$

$$E_i = \sqrt{J^2 \langle S_{i-1} + S_{i+1} \rangle^2 / 4 + H_x^2} \quad (5.29)$$

これによりモデルは1次元のまま、計算量は従来のQMCよりも格段に少なくてすむ。すなわち、1次元イジングモデルをMCで計算するのと同じ計算量となる。その計算手順は、

1. スピンの番号 i を選ぶ。
2. $[0, 1]$ の一様乱数 r を発生し、

$$r < \frac{1}{2} \left[1 + \frac{J \langle S_{i-1} + S_{i+1} \rangle / 2}{E_i} \tanh(\beta E_i) \right] \quad (5.30)$$

ならば $S_i = +1$, そうでなければ $S_i = -1$ とする。

である。MCと比較して、確率の大きさが異なっている事が注目すべき点である。しかしながら、これを繰り返し適用してもTIモデルのエネルギーの最小値が求まる保証はないため、徐々に量子揺らぎの項 H_x を小さくしていくという方法を適用する。これにより、TIモデルではなくイジングモデルのエネルギーの最小値が求まるのである。

次に、このMFAを用いたQMCの実験を行う。\$N\$個のスピンの1次元TIモデルに対して、片方の端を固定端とし、もう片方の端を開放端とするような境界条件を課す。また初期状態は、\$S_1 = 1, S_2 = S_3 = \dots = S_N = -1\$とし、\$S_1\$を固定する。ケットベクトルで書くと、\$|\uparrow, \downarrow, \dots, \downarrow\rangle\$である。この状態は\$S_1\$を反転すれば\$H_1\$の基底状態であるが、\$S_1\$が固定なので、基底状態に達するためには残りの\$N-1\$個のスピンを全て反転させなければならない。これは1種のだまし問題とも言える。また、比較のために1次元イジングモデルをMCでも計算を行う。最終的には、交換相互作用項\$H_1\$の最小値を求めたいのであるから、\$t \to \infty\$で量子項\$H_2 \to 0\$とすればよい。これを実現する方法は前述のSAに類似しているが、量子スケジュールとして、\$H_x(t) = H_{x0} \times 0.9^t\$とすれば、更新回数が十分大きいところでハミルトニアン\$H\$の期待値\$\langle H \rangle\$が\$\langle H_1 \rangle\$に近づく。よって、ここでは簡単のため、エネルギーの計算においては、量子項の期待値\$\langle H_2 \rangle\$は落す。温度スケジュールに関しては前節と同様、指数関数的な減少、すなわち初期温度を\$T_0\$とし、\$T(t) = T_0 \times 0.9^t\$とする。また、比較的速く緩和するアルゴリズムとして知られるcheckerboard updationを行う。これは、1から\$N\$までのスピンを交互に白と黒に塗り分け、白、すなわち奇数番目のスピンを全て反転させてから、次に黒、すなわち偶数番目のスピンを反転させるという方法である。

MFAを用いたQMCによるエネルギーの緩和シミュレーションの実験結果を、図5.2に、更新回数を対数的に表示したものと、縦軸を対数的に表示したものを示す。また、パラメータはキャプションの通りである。\$H_{x0} = 0\$の曲線は、量子項がない場合の緩和であり、これはMCによってイジングモデルのシミュレーションを行ったのと全く同じである。左図から、量子項の初期値\$H_{x0}\$を大きくとることにより、エネルギーの極大値が大きくなっている。これは、量子揺らぎを加えることによる効果である。また、右図から、\$H_{x0}\$が大きい程、わずかながら緩和速度が速いことが分かる。これは一旦エネルギーが高くなった方が、局所解から抜け出す可能性が大きいためである。

もう一つのシミュレーション結果を、図5.3に示す。この場合は初期温度を\$T_0\$を0、および1としたものである。左図から、量子揺らぎのみの場合よりも、量子揺らぎ+温度揺らぎの場合の方がエネルギーは高くなっている。しかしながら、予想に反して、右図から、温度揺らぎがない方が緩和時間がわずかに短くなっているのである。これは図5.2で考察したような状況があてはまらない。温度揺らぎがない方が、エネルギーは低いにもかかわらず、局所解から抜け出す割合が大きいという事を示している。

このからくりはおそらく、温度揺らぎと量子揺らぎの振舞いの違いに関係していると思われる。この両者は、量子揺らぎを0とし、高温から絶対零度までゆっくり冷やすスケジュールと、温度を0とし、量子揺らぎを徐々に小さくしていくスケジュールの、線形応答から、非線形応答への移行として把握することが出来るが、異なる点は、その確率の大きさであり、式で表すと、式(5.28)において、絶対零度におけるスピンの期待値と、量子揺らぎが0の場合のスピンの期待値を簡単な式で比較すると、

$$\frac{|S|}{\sqrt{S^2 + x^2}} < \tanh(|S|/x) \quad (5.31)$$

となる。ただし\$S\$は任意のスピンを表し、左辺の\$x\$は\$H_x\$、右辺の\$x\$は温度である。モンテカルロ法により、両辺をそれぞれ\$[0, 1]\$の乱数に投影してみると、左辺の方がスピンの確定せず、揺らぐ割合が多いのである。従って、量子揺らぎと、温度揺らぎを同じスケジュールで

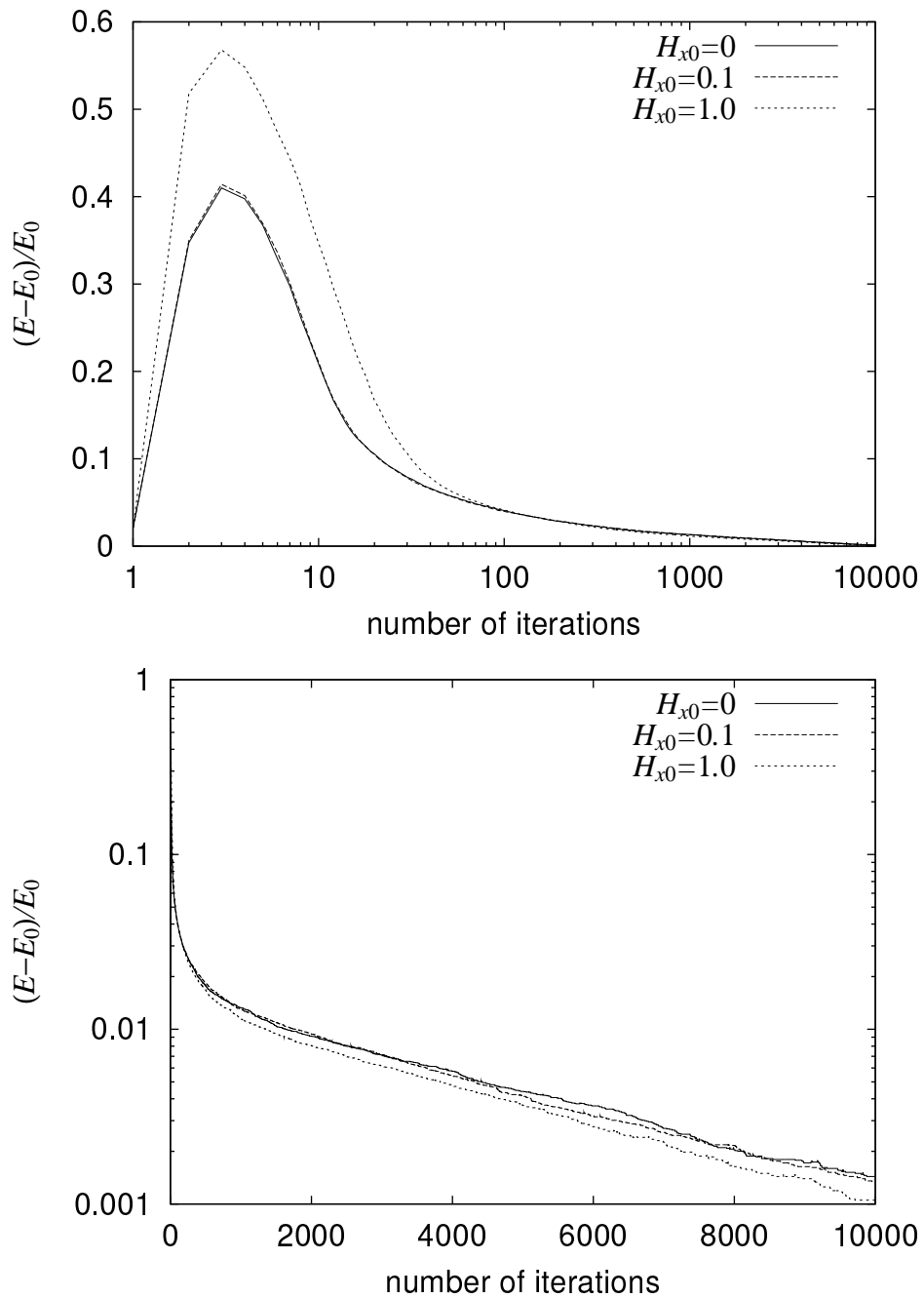


図 5.2: 平均場近似(MFA)を用いた量子モンテカルロ法(QMC)によるエネルギーの緩和. パラメータは, $N = 100$, $J = 1$, $T(t) = T_0 \times 0.9^t$, $T_0 = 1$, $H_x(t) = H_{x0} \times 0.9^t$, $N_{\text{sample}} = 1000$, $E_0 = -(N-1)J$. H_{x0} の値はそれぞれ, 0, 0.1, 1.0.

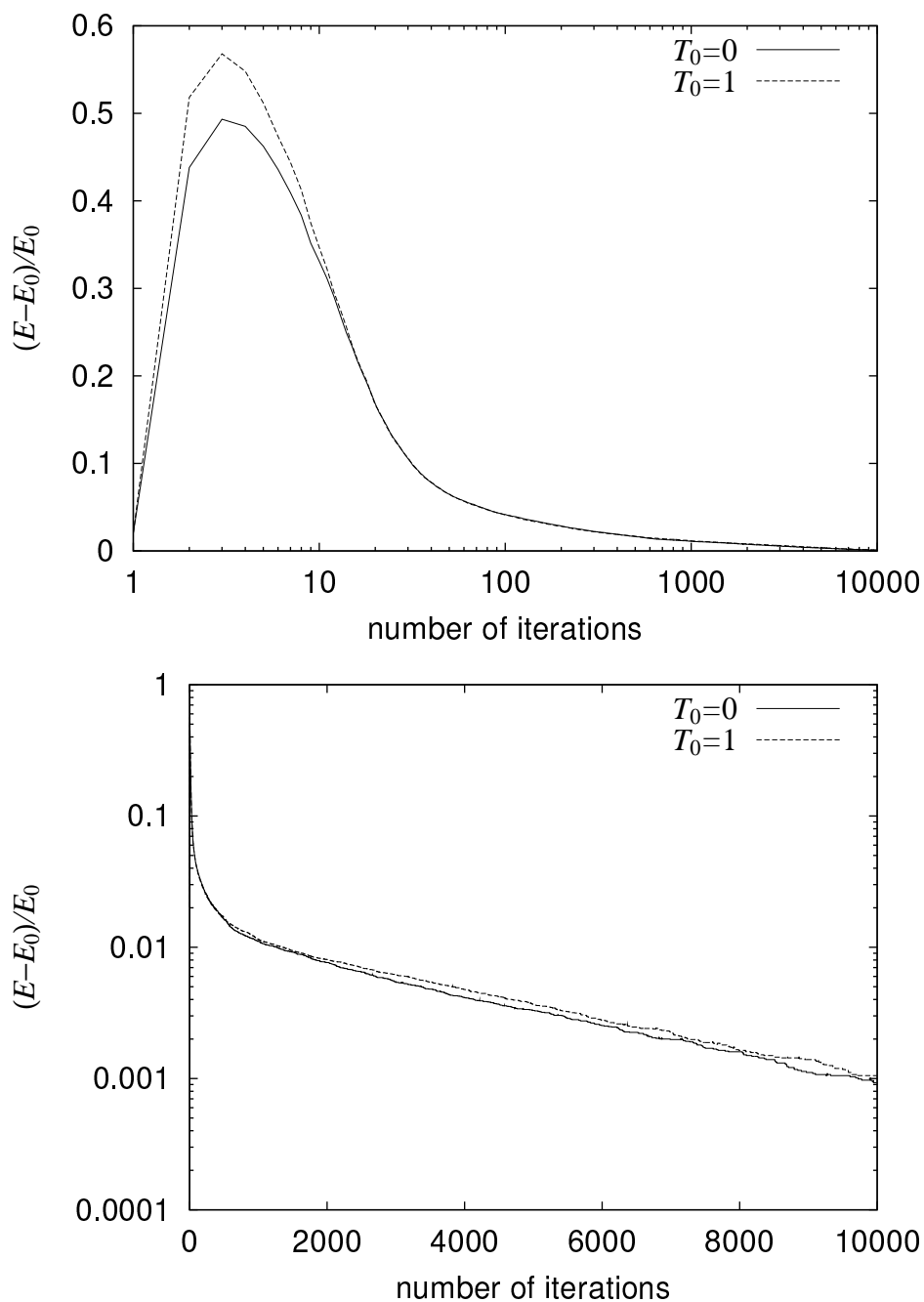


図 5.3: 平均場近似(MFA)を用いた量子モンテカルロ法(QMC)によるエネルギーの緩和. パラメータは, $N = 100$, $J = 1$, $T(t) = T_0 \times 0.9^t$, $H_x(t) = H_{x0} \times 0.9^t$, $H_{x0} = 1$, $N_{\text{sample}} = 1000$, $E_0 = -(N-1)J$. T_0 の値はそれぞれ, 0, 1.0.

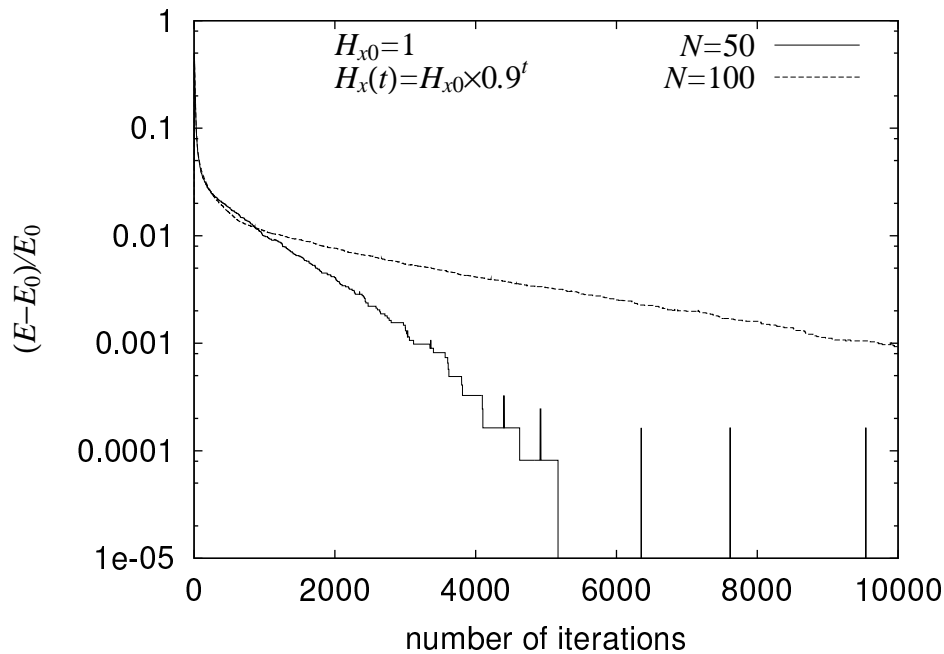


図 5.4: 平均場近似 (MFA) を用いた量子モンテカルロ法 (QMC) によるエネルギーの緩和. パラメータは, $J = 1$, $T_0 = 0$, $H_x(t) = H_{x0} \times 0.9^t$, $H_{x0} = 1$, $E_0 = -(N - 1)J$. サイズ N はそれぞれ, 50, 100.

与えて比較すると, 各スピンの揺らぐ割合が, 量子揺らぎで与えた方が多いため, 局所解から抜け出す割合が多いのである.

最後に, サイズを変化した場合の計算結果を示す. 図 5.4 は, $N = 50$, および 100 の場合のエネルギーの緩和である. 温度は 0, 量子揺らぎのスケジュールは前述のものと同様である. 図より, 片対数で直線的であるから, $t \rightarrow \infty$ で指数関数的な減衰 $e^{-t/\tau}$ であると言える. 図の傾きからこの緩和時間 τ を見積もると, $N = 50$ の時, $\tau \sim 10^3$, $N = 100$ の時, $\tau \sim 4 \times 10^3$ となっている.

5.4 メトロポリス法の関数最適化問題への適用

5.4.1 アルゴリズム

MC を一般の問題にも適用できるように拡張したものがメトロポリス法 (MP: Metropolis Method) である. これは, ある状態における関数値と摂動を加えた後の関数値の差を, 温度によって重み付けしたもののみを用いるため, どのような問題にも非常に簡単に適用できるという利点がある. また, MP を適用した場合の状態の分布が, MC の場合と同様, 熱平衡状態で Gibbs 分布に等しくなることも証明されている.

強磁性イジングモデルにおける MP の計算手順は次のように形式化されている.

1. スピンの番号 i を選ぶ.

2. スピン S_i が符号を変えたときのエネルギー変化

$$\Delta E = 2JS_i(S_{i-1} + S_{i+1}) \quad (5.32)$$

を計算する.

3. $[0, 1]$ の一様乱数 r を発生し,

$$r < \exp(-\Delta E/T) \quad (5.33)$$

ならば $S_i \rightarrow -S_i$ とする.

である. ΔE が負の場合は変化後のエネルギーの方が小さいということを表し, この場合は必ず更新する. また, ΔE が正の場合は変化後のエネルギーの方が大きく, この場合は確率 $\exp(-\Delta E/T)$ で更新するのである.

この方法を一般の問題に適用する方法は, 問題の定義変数の一部を選び, 対立する表現方法に変更するとき, 関数値の変化量を ΔE とし, 確率

$$P(\Delta E) = \exp(-\Delta E/T) \quad (5.34)$$

によって更新するかどうかを決定する. ここでは, GA との比較を目的としているため, 問題の定義変数の表現方法は GA と同様, Gray coding とする. 再び, 計算手順をまとめると,

1. Gray coding された遺伝子の番号 i をランダムに選ぶ.
2. 遺伝子 γ_i が反転したときの適応度の変化量 ΔE を計算する.
3. $[0, 1]$ の一様乱数 r を発生し, $r < P(\Delta E)$ ならば γ_i を反転する.

となる. これを繰り返し, 終了条件を満たしていれば終了する. このアルゴリズムは個体数 1 の GA と等価であり, 個体に対する演算が突然変異のみであるアルゴリズムに相当する. また, 突然変異率は状態によって動的に変化するものであるとみなせる.

次に, MC から MP への拡張方法にならって, QMC から MP への拡張を試みる. MC と QMC の異なる点は, 問題変数に加えられる摂動を受け入れるか否かの, 確率の大きさのみである. そこで, 関数値の変化量を ΔE とし, 温度が 0 で, 量子揺らぎ H_x のみが存在する場合の確率を,

$$P(\Delta E) = (\sqrt{\Delta E^2 + H_x^2} - \Delta E)/H_x \quad (5.35)$$

で与える. これについては, ΔE が大きいところで $1/\Delta E$ の振る舞いをするような裾野の広い分布であり, 式 (5.31) を用いて導かれる. この確率を用いて MP の計算手順をまとめると,

1. Gray coding された遺伝子の番号 i をランダムに選ぶ.
2. 遺伝子 γ_i が反転したときの適応度の変化量 ΔE を計算する.
3. $[0, 1]$ の一様乱数 r を発生し, $r < P(\Delta E)$ ならば γ_i を反転する.

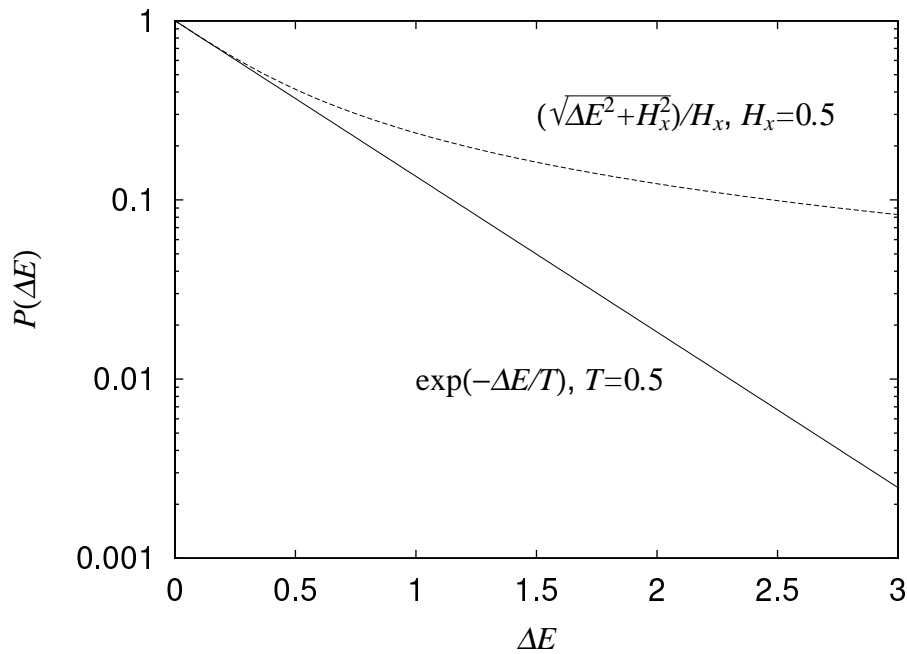


図 5.5: 温度揺らぎによる分布と量子揺らぎによる分布

である。これを、終了条件を満たすまで繰り返す。

以上が、MC、および QMC の MP を用いた計算法であるが、両者共に個体数 1 の GA と等価であり、突然変異のみで解を探索していくアルゴリズムであることが分かる。また、MC、および QMC の違いは局所的な確率分布のみである。MC の場合は指数分布であるのに対し、QMC の場合は裾野の広がった分布である。これは量子効果に特有のものであり、式 (5.35) と同様の関係式、

$$(\sqrt{\Delta E^2 + x^2} - \Delta E)/x > \exp(-\Delta E/x) \quad (5.36)$$

から、温度パラメータと量子パラメータが同じ大きさのとき、温度揺らぎよりも量子揺らぎの方が大きいことが確認される。図 5.5 は温度パラメータと量子パラメータの大きさが 0.5 のときの分布である。

5.4.2 実験と結果

MP によって拡張された MC、および QMC を用いて関数最適化を行う。まず、目的関数については、第 2 章で定義したベンチマーク 9 関数の 5 次元版を用い、遺伝子長も同様のものを用いる。評価回数は 10,000 回とし、独立に 300 回の試行によって、成功率、ENES の統計量を求める。また、MC の場合は SA を併用するため、パラメータは、初期温度 $T_0 = 1$ 、アニーリングスケジュールは、 $T(t) = T_0 \times 0.99^t$ とした。QMC の場合も同様に、初期量子揺らぎを $H_{x0} = 1$ 、アニーリングスケジュールは、 $H_x(t) = H_{x0} \times 0.99^t$ とした。すなわち、MC における温度のアニーリングスケジュールと同じ、指数関数的な減少である。

実験結果は表 5.1 のようになった。Sp, Rs では MC, QMC 共に成功率は 100% となり、Ds,

表 5.1: モンテカルロ法と量子モンテカルロ法による成功率と ENES

Func	MC		QMC	
	SR(%)	ENES	SR(%)	ENES
Sp5	100	1644	100	1729
Ds5	26	8788	25	8906
Rs5	100	1652	100	1719
Rd5	26.67	8838	28.67	8512
Ra5	0	-	0	-
Gr5	0	-	0	-
Mi5	8	968	7.67	1057
Sh5	1.33	641	0.67	870
La5	0.67	1076	0.33	269

表 5.2: ICEO5 関数に対するモンテカルロ法, 量子モンテカルロ法, および PfGA による成功率と ENES

Func	MC		QMC		PfGA	
	SR(%)	ENES	SR(%)	ENES	SR(%)	ENES
Sp5	100	1644	100	1729	100	4002
Gr5	0	-	0	-	0.33	9670
Sh5	1.33	641	0.67	870	7.67	1758
Mi5	8	968	7.67	1057	29.67	3170
La5	0.67	1076	0.33	269	38	4159

Rd では 25~28%前後という結果となった。これらは単峰性関数であるが, Ra から La まで
は多峰性関数であり, 成功率が極端に低いことが分かる。Ra, Gr については成功率は 0%で
あり, 大域的な探索にはあまり向いていないという傾向にある。ENES については, MC で
は, Mi, Sh で 1000 以下となり, QMC では La に対する収束性能が際だっている。

5.5 考察: モンテカルロ法と遺伝的アルゴリズムとの比較

実験結果から, MC と QMC の性能はあまり違いがないことが分かる。では, MC(QMC)
と GA とを比較した場合, その違いはどれほどのものとなるであろうか。表 5.2 は, MC,
QMC, および PfGA の, ICEO5 関数に対する成功率と ENES を, 前節の結果と第 2 章の結
果をもとにまとめ直したものである。表から, MC, QMC は収束性能に優れているのに対
し, PfGA は成功率に優れていることが分かる。

アルゴリズムを対比させると, GA では, 交叉, 突然変異, および選択の 3 つの演算子を用
いているが, MC(QMC) では, 温度揺らぎや量子揺らぎに従って状態を遷移させるかどう

かという遷移則のみを用いている。また、MC(QMC)では個体数は1であるため、探索途中で出現する最良解を保存するエリート戦略が使えない。温度(量子)パラメータは局所解から抜け出すための揺らぎの度合であり、収束を速くするための改良を行うには、この温度(量子揺らぎ)を調節するか、あるいは、アニーリングスケジュールを設定するなどの職人芸的な方法が求められる。しかしながら、成功率の向上にはGAの持つ多様性が必要であり、MC(QMC)による大域的探索はやはり非常に困難であると言わざるを得ない。

GAにおける主要演算子は交叉であると言われているが、基本的には交叉と選択が重要であると考えられる。交叉は個体を増やす演算子であり、選択は個体を減らす演算子である。個体数の増減を許すことにより、さまざまな改良(戦略)を施すことができるのである。ではGAで、個体数を問題に応じて最適にするような機構とはどのようなものなのであろうか。我々は第2章において、PfGAの選択則を試行錯誤により決定したが、これを次のように別の側面から考察してみる。

物理系では、粒子数の一定でない系は「流れ」のある系とみなされ、系の外部と、エネルギーと粒子の両方を交換するモデルが用いられる。またこのような系はグランドカノニカル集団(grand canonical ensemble)と呼ばれ、熱力学的な特性は、次のGibbs分布(グランドカノニカル分布)

$$\rho = \frac{1}{Z} \exp[-\beta(H - \mu N)] \quad (5.37)$$

$$Z = \text{Tr} \exp[-\beta(H - \mu N)] \quad (5.38)$$

を用いて計算される。 H は系のハミルトニアン(エネルギー関数)、 N は全粒子数である。また、 μ は化学ポテンシャル(chemical potential)と呼ばれる量で、例えばバケツに入っている水の高さに相当するものである。 Z は大分配関数(grand partition function)である。このような系は一般に熱平衡から遠く離れた非平衡状態にあり、定常状態では、GibbsとHelmholtzによって与えられた熱力学的ポテンシャル、

$$\begin{aligned} \Omega &= \langle H \rangle - \mu \langle N \rangle - TS \\ &= F - \mu \langle N \rangle \end{aligned} \quad (5.39)$$

が最小となっている。ただし、 S はエントロピー、 F はHelmholtzの自由エネルギーである。非平衡系の解析が困難である理由の一つが、定温、定積の下での熱力学ポテンシャルの変化量 $d\Omega$ について、2通りの解釈があることである。その2つとは、パラメータを μ ととるか、あるいは N ととるかという問題であり、モデルによって使い分けなければならない。

非平衡系における定常状態では、カノニカル統計における熱平衡状態と区別して、「流れ平衡状態」と呼ぶこともある。文献[41],[42]などでは、量子ドットを流れる定常電流の期待値を求めるための方法として次のような計算を行っている。まず、電流一定の条件より、長時間平均で、入ってくる電子の合計と、出ていく電子の合計が等しいとおくことにより μ を決定し、この μ を Ω に代入することにより、 N の関数としての $\Omega(N)$ を得ることができる。この $\Omega(N)$ を最小にする N が量子ドット内の電子数であり、これら2つの変数 μ と N を密度行列(5.37)に代入し、この密度行列から、更新された電流の期待値を求める。これを何回も繰り返すと定常状態における正しい電流の期待値が得られる。

これに類似した研究として、Bergらによるmulticanonical algorithmなどがある[43]。式(5.37)において、 $-\mu N$ の部分が H の関数 $f(H)$ となっている計算方法であり、1次の相転

移などに適応され、成果を上げている。このように、分布の形を修正したものを用いる発想の根底には、やはり非平衡系の概念があり、この場合は化学ポテンシャルや粒子数のようなものの代わりに、ある関数 $f(H)$ を用いて強制的に揺らぎを加えているのである。

一方、非平衡系のモデルとして有名なものに、パー・バックらによる砂山モデル (sandpile model) という雪崩をシミュレートするモデルがある [44]。これは、与えられた空間上に砂をランダムに落していき、ある領域の砂の高さがしきい値を越えると、砂の増分はその隣の領域に移動し、端では隣がないので下に落ちていくという簡単なモデルである。このモデルでは、ある一定の山の高さと傾きを保っている状態は臨界状態と呼ばれ、砂を蓄積していく状態と、山が雪崩によって崩壊する状態の中間の状態を指しており、この状態を自らが保とうとしていることから、彼らは「自己組織的臨界状態」と呼んだ。また、連鎖反応によって、雪崩の規模と頻度の関係がべき乗法則に従っていることも示している。物理的な解釈をするならば、山の高さと傾きがまさに N と μ に相当し、臨界状態とは、熱力学ポテンシャルが最小になっている状態であると捉えることができる。

では、砂山モデルなどの非平衡系を最適化に応用することは可能であろうか。これに対しては、非平衡系における定常状態が、熱力学ポテンシャル Ω が最小になっている状態であるため、 Ω を目的関数とするように問題を coding しなければならない。次の作業としては、現象論的なマスター方程式を用いて粒子1つの運動方程式を立てることである。この時の係数は、時間が十分経た後の分布が Gibbs 分布 (5.37) となるように決定する。これにより得られるダイナミクスを多くの粒子に対して適用し、平均操作をすれば、マクロな量の時間発展が得られるであろう。また、時間が十分経過した後の熱力学ポテンシャルの期待値は最小値をとっているはずである。このような計算法の応用として、温度をゆっくり 0 に近づけていく方法と、化学ポテンシャルをゆっくり 0 に近づけていく方法を併用することにすれば、時間が十分経た後の内部エネルギー $\langle H \rangle$ は最小となっている。問題は内部エネルギーの項を目的関数とするような coding の仕方を考えることに集約されるであろう。

以上が非平衡系における MC の適用方法の概略である。このアルゴリズムにおいて、粒子1つの運動方程式を、GA における個体群の運動方程式にまで拡張することができれば、より最適な交叉、選択方法を理論的に得ることができると可能性がある。結局、GA の理論解析には非平衡多体系の概念が必要となるのである。

5.6 まとめ

本章では、統計力学的計算法と進化的計算法との関係を明らかにする目的で、モンテカルロ法と遺伝的アルゴリズムとの比較を行った。モンテカルロ法については、一般の関数にも適用できるようにメトロポリス法に拡張し、また、量子モンテカルロ法についてもメトロポリス法に拡張し、同様に比較を行った。

モンテカルロ法と量子モンテカルロ法の比較では、目的関数がイジングモデルのエネルギーの場合にはわずかながら違いが現れ、量子モンテカルロ法の方が収束性能に優れていることが確認されたが、一般の関数に対しては、違いはあまり現れなかった。

一方、モンテカルロ法と遺伝的アルゴリズムの比較に関しては、モンテカルロ法は収束性能に優れているのに対し、遺伝的アルゴリズムは大域的探索性能に優れていることが分かった。モンテカルロ法は個体数1個の遺伝的アルゴリズムと等価なアルゴリズムであるため、

大域的探索には向かないという結論に達した。

モンテカルロ法に代表される統計力学的計算法は、そのアルゴリズムの構造上、進化的計算法に含まれるものである。このことは、統計力学的計算法が平衡系のモデルから出発していることが原因である。進化的計算法は生物の進化や生態系のモデルから出発したアルゴリズムであり、生物や生態系そのものは非平衡多体系であるからである。

第6章 考察

6.1 理論的考察

GAの理論解析に関してはさまざまな試みがなされている。目的関数を単純化することで、個体の分布のダイナミクスを解析的に表現しようとするものである。しかしながら、やはり困難であることは事実であるので、目的関数を想定することはやめ、遺伝子を実数変数に decode した時の振舞を統計的に追ってみる。ただしここでは、突然変異のような簡単な演算に対する解析を中心に話を進める。

6.1.1 突然変異に対する理論解析

GAによって実数関数最適化を行う場合、個体の表現形は実数変数である。では、遺伝子の突然変異は、実数に変換した時、どれほど変化するものなのであろうか。そこで、ある Gray coding された遺伝子を突然変異率 p_m で突然変異させ、これを実数変数に decode することを考える。

任意の突然変異率についての解析は困難であるため、まず遺伝子長の逆数を単位として用い、1 bit だけ反転させた場合を考える。長さ n の Gray coding された任意の遺伝子 γ に対し、その j 番目の遺伝子 γ_j を反転させる演算子を Q_j とする。すなわち、

$$Q_j \gamma_i = \delta_{ij}(1 - \gamma_i) \quad (6.1)$$

と書ける。binary coding された bit 列 β に対しては、式 (1.3) を用いて、binary coding を Gray coding に変換して演算を行う。

$$\begin{aligned} Q_j \beta_i &= Q_j \left[\left(\sum_{k=1}^i \gamma_k \right) \pmod{2} \right] \\ &= Q_j [(\gamma_1 + \gamma_2 + \cdots + \gamma_i) \pmod{2}] \end{aligned}$$

ここで、 $i < j$ の時は Q_j による変化はないが、 $i \geq j$ の時は、

$$\begin{aligned} Q_j \beta_i &= (\gamma_1 + \gamma_2 + \cdots + 1 - \gamma_j + \cdots + \gamma_i) \pmod{2} \\ &= 1 - \beta_i \end{aligned}$$

となる。これは β_i を反転させることに相当する。以上をまとめて書くと、

$$Q_j \beta_i = \begin{cases} \beta_i & (i < j) \\ 1 - \beta_i & (i \geq j) \end{cases} \quad (6.2)$$

となる. 次に, j 番目の bit を反転した時の x の変化量 d_j を, $d_j = x - Q_j x$ と定義し, その期待値を $j = 1$ から n までで平均したものとする. x の定義式 (1.1) を用いて,

$$\begin{aligned} d_j &= x - Q_j x \\ &= \frac{L}{2^n - 1} \sum_{i=j}^n 2^{n-i} (2\beta_i - 1) \end{aligned} \quad (6.3)$$

$$\begin{aligned} \langle d \rangle &= \frac{1}{n} \sum_{j=1}^n (x - Q_j x) \\ &= \frac{1}{n} \sum_{j=1}^n \frac{L}{2^n - 1} \sum_{i=j}^n 2^{n-i} (2\beta_i - 1) \end{aligned} \quad (6.4)$$

となる. ただし, L は定義域の長さである. $(2\beta_i - 1)$ については, β_i が取り得る全ての状態で平均すると 0 となるので, $\langle d \rangle = 0$ が得られる. 次に d の分散を求めると,

$$\begin{aligned} \langle \Delta d^2 \rangle &= \langle d^2 \rangle \\ &= \frac{1}{n} \sum_{k=1}^n (x - Q_k x)^2 \\ &= \frac{1}{n} \sum_{k=1}^n \left[\frac{L}{2^n - 1} \sum_{i=k}^n 2^{n-i} (2\beta_i - 1) \right]^2 \\ &= \frac{1}{n} \left(\frac{L}{2^n - 1} \right)^2 \sum_{k=1}^n \sum_{i=k}^n \sum_{j=k}^n 2^{2n-i-j} (2\beta_i - 1) (2\beta_j - 1) \\ &\sim \frac{L^2}{n} \sum_{k=1}^n \sum_{i=k}^n \sum_{j=k}^n \left(\frac{1}{2} \right)^{i+j} S_i S_j \end{aligned} \quad (6.5)$$

のようになる. この式で, $S_i = 2\beta_i - 1$ の項は ± 1 をとるので, 反磁性イジングモデルのエネルギーを表している. この値は, $i = j$ の時の寄与が残り, $i \neq j$ の時は i と j の対称性のために消える. よって,

$$\begin{aligned} \langle \Delta d^2 \rangle &= \frac{L^2}{n} \sum_{k=1}^n \sum_{i=k}^n \left(\frac{1}{2} \right)^{2i} \\ &= \frac{L^2}{n} \sum_{k=1}^n \frac{4}{3} \left[\left(\frac{1}{4} \right)^k - \left(\frac{1}{4} \right)^{n+1} \right] \\ &= \frac{4L^2}{3n} \left[\frac{1}{3} - \frac{1}{3} \left(\frac{1}{4} \right)^n - n \left(\frac{1}{4} \right)^{n+1} \right] \\ &\sim \frac{4}{9} L^2 \frac{1}{n} \quad (n \gg 1) \end{aligned} \quad (6.6)$$

が得られる. $1/n$ は 1 bit 反転したときの突然変異率 p_m に等しい. 2 bit 反転したときも同様にして $d_{ij} = x - Q_i Q_j x$ から導いていくと,

$$\langle \Delta d^2 \rangle \sim \frac{4}{9} L^2 \frac{2}{n} \quad (6.7)$$

となる。 $2/n$ は 2 bit 反転したときの突然変異率である。 これらより、任意の突然変異率 p_m ($\ll 1$) に対して、

$$\langle \Delta d^2 \rangle \sim \frac{4}{9} L^2 p_m \quad (6.8)$$

$$\langle \Delta d \rangle \sim \frac{2}{3} L \sqrt{p_m} \quad (6.9)$$

が得られる。

また、同様の手順で、 d の高次のモーメントを求めることができ、その結果は、

$$\langle d^{2w} \rangle \sim \frac{2^{2w-2} - 1}{2^{2w} - 1} \left(\frac{4}{3} \right)^w L^{2w} p_m \quad (6.10)$$

となる。

次に、実験的にはどのような振舞を示すかを調べる。 遺伝子長 n のある個体に対し、突然変異率を $p_m = 1/n$ として個体に突然変異を適用した時の移動距離を、分布として累計をとったものが図 6.1 である。 ただし、 $n = 24$ 、定義域の大きさは $L = 10$ である。

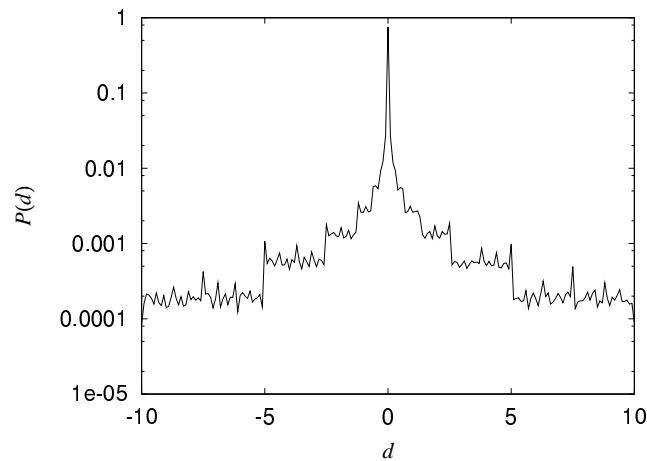


図 6.1: 突然変異率 $p_m = 1/n$ によって摂動を受けた個体の分布 ($n = 24$)

図から、分布の裾野が広範囲にわたって広がり、また、細かな極大値を持つ複雑な構造となっている。 この分布は、ガウス雑音のような分布とは異質のものであることが分かる。 また、式 (6.9) より、この場合の揺らぎはおよそ 1.36 と計算されるので、図示される結果が妥当なものであることが理解される。

以上の結果から、局所探索領域に対する突然変異率の、定量的な設計指針を与える可能性が示された。 すなわち、多峰性関数の谷から谷 (最大値探索の場合は山から山) への遷移幅を関数に応じて設計すればよいことになる。

さらに、突然変異による裾野の広がった分布は、第 5 章で述べた量子揺らぎ (5.35) に非常によく似ているということも興味深い。 この点については、量子揺らぎ自体が coding の仕方に無関係に存在しているため、単純に比較することはできないが、Gray coding された遺伝子に対する突然変異の性質としては、温度揺らぎよりも量子揺らぎの方が近いという可能性もある。 これについては今後さらなる解析が必要であろう。

6.1.2 交叉に関する考察

交叉に関しては、前述のような枠組で解析できるわけではない。それは、遺伝子の bit 相関が関係してくるからである。相関が大きい場合はスキーマが保存され、小さい場合は、各遺伝子がランダムに個別の振舞をすると考えられる。よって、解析的には非常に複雑になるため、実験的に振舞を確認することにする。

親個体 P_1, P_2 を用い、一様交叉によって子個体 C_1, C_2 が発生したとする。親個体のハミング距離を $H = |P_1 - P_2|$ とすると、子個体どうしのハミング距離 $|C_1 - C_2|$ も H に等しくなる。また、一様交叉であるから、親子間のハミング距離は平均して $H/2$ となる。

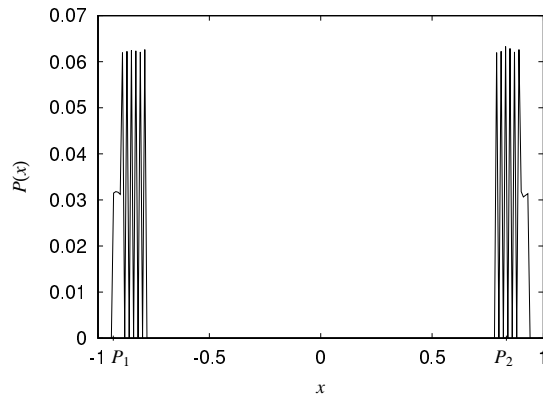


図 6.2: $H = 8$ の時の子個体の分布

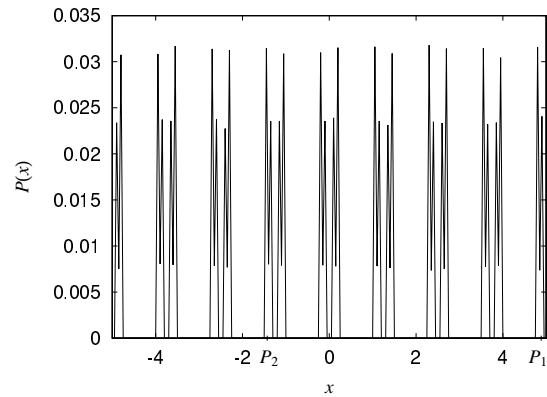


図 6.3: $H = 16$ の時の子個体の分布

図 6.2, 6.3 は、親どうしのハミング距離を $H = 8, 16$ とした時の、一様交叉により発生する子個体の分布である。図中の P_1, P_2 は親個体の座標である。また、個体の遺伝子長は 24 bit であり、定義域は $[-5, 5]$ である。図より、親どうしのハミング距離が小さい場合は、子個体はそれぞれの親の近傍に発生しやすくなり、親どうしのハミング距離が大きい場合は、子個体は全探索空間に広く分散することが分かる。SSGA のような集団ベースの GA では、探索が進むにつれ、親として選ばれる個体のハミング距離が徐々に小さくなり、探索後期では局所探索に近くなるような状況となる。交叉が本質的に重要であることの理由がこれである。言い替えると、個体集団が探索状況に対して適応的に変化していると解釈することもできる。

6.2 評価基準に対する考察

本論文で採用した評価基準は主に、成功率 (SR), および平均評価回数 (ENES) である. これらを用いることにより, どのような GA が, どのような問題に対して有効であるかという詳細な情報を系統的に知ることができた. この2つの基準はそれぞれ独立ではなく, 相互に深く関係している. このことを, 評価回数の分布を計算することにより調べてみる.

まず, 図 6.4 は, 5次元 Sphere model に対して, PfGA を用いて計算を行った場合の, 評価回数 t の分布である. ただし, 試行回数は 10000 回であり, したがってサンプル数は 10000 個である. 図から分かるように, 終了条件である $T = 10,000$ の中に分布が収まっており, この場合の成功率が 100% であることと, ENES がおよそ 4000 であること (表 2.2 参照) を満足している.

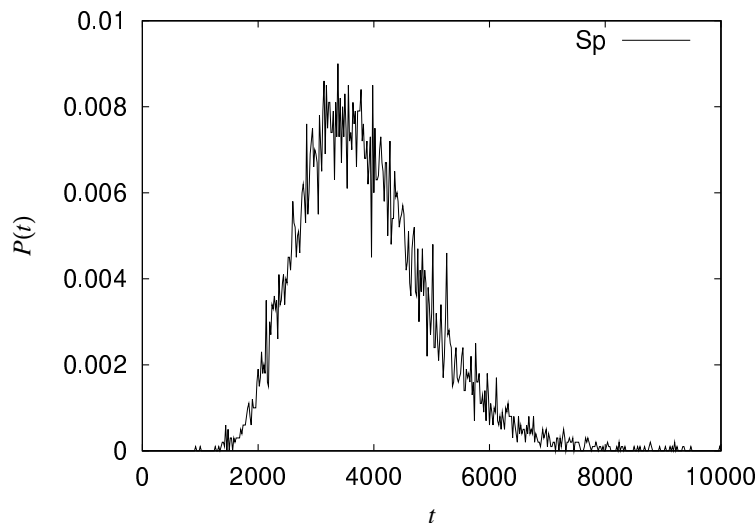


図 6.4: 評価回数の分布 (5-d Sphere model, PfGA)

これを解析的に表現してみると, 次のようになる. 評価回数の分布を $P(t)$ とし, 評価回数の上限を T とすると, 成功率 $R(T)$, および ENES $E(T)$ は,

$$R(T) = \sum_{t=1}^T P(t) \quad (6.11)$$

$$E(T) = \frac{\sum_{t=1}^T tP(t)}{R(T)} \quad (6.12)$$

と表される. ここで, ENES は成功試行でのみ平均操作しているため, 成功率で規格化しているのである. この式より, 成功率と ENES は, ある分布 $P(t)$ を介して密接に関係していることが分かる. その他の関数に対しても同様の関係が成り立ち, また, その他の GA による探索においても, 同様の関係が成り立つ. ただし, 分布 $P(t)$ は目的関数によって異なり, また, その他の GA による探索においても異なる. 例えば局所解に陥る可能性が高い関数に対しては, t が大きいところで裾野が広がっているような分布となることが予想される.

図 6.5 は 5 次元 Langerman's function に対して, PfGA を用いて探索を行った場合の評価回数分布である. この場合の成功率が 38% であるから, 図に示される分布の積分面積のさらに 2 倍が, $t = 10,000$ よりも右の領域に存在すると考えられる. これに対し, 図 6.6 は同じ関数に対して, SSGA を用いて探索を行った場合の分布である. この場合の成功率は 90% を越えているため, 分布のほとんどの領域が図中に収まっている. これらの結果から, アルゴリズムが異なれば, 分布の形も異なってくるのが分かる.

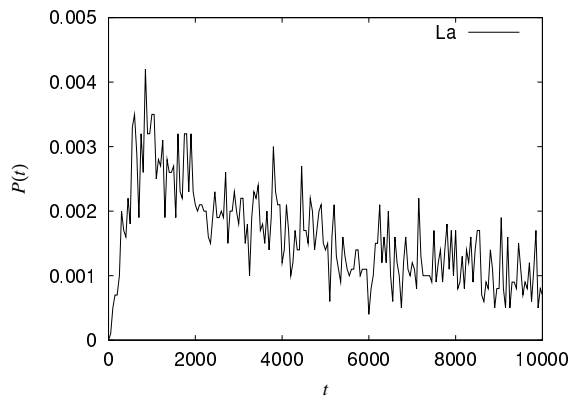


図 6.5: 評価回数の分布 (5-d Langerman's function, PfGA)

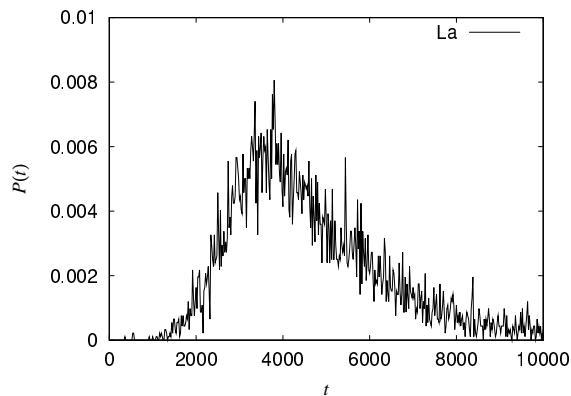


図 6.6: 評価回数の分布 (5-d Langerman's function, SSGA)

異なるアルゴリズムのもう一つの例として, 第 3 章で述べた並列 GA ではどのような振舞となるかを考えてみる. 並列処理を行う際, 評価回数の上限を T とし, 並列度を n とすると, 1 タスク, すなわち 1 局所集団当たりの評価回数の上限は T/n である. よって, 1 局所集団が解を見つける確率は, 局所集団間の相互作用がない場合, すなわち移民がない場合,

$$R_0 = \sum_{t=1}^{T/n} P(t) \quad (6.13)$$

である. このような局所集団が n 個あるので, 全体としての成功率は, 1 から, 全ての局所集団で探索が失敗する確率を引けばよい. すると,

$$R_n = 1 - (1 - R_0)^n \quad (6.14)$$

となる. この並列探索による成功率 R_n がもとの成功率 $R(T)$ よりも大きくなれば, 並列の効果が現れたと解釈できる. 上式により, 並列 GA の, 移民がない場合の成功率が $R(T)$ より大きくなるためには,

$$1 - \sqrt[n]{1 - R(T)} < R_0 < R(T) \quad (6.15)$$

であればよい. ただし右辺は R_0 の定義より明らかである. この式からただちに,

$$n > \frac{\log [1 - R(T)]}{\log (1 - R_0)} \quad (6.16)$$

となり、分布関数を精度よく測定することにより n の下限が決定される。これは移民がない場合の関係式であるため、並列化によって局所集団間での個体の移民をも考慮すると、分布は変化すると考えられる。したがって、アルゴリズムの改良次第では、並列化の効果が現れる可能性はさらに大きくなるのである。

第7章 結論

7.1 研究成果

本研究における研究成果について、各章ごとにまとめる。

7.1.1 パラメータフリー遺伝的アルゴリズム

第2章では、汎用的な探索アルゴリズムを開発することを目的として、パラメータフリー遺伝的アルゴリズム (PfGA) を考案し、その性能評価を行った。PfGA は、family 内の個体の適応度の大小に応じて 1~3 個体を選択し、大局的な探索と局所的な探索のバランスを動的に取りながら最適解を見つけ出していくアルゴリズムである。このアルゴリズムのパフォーマンスを調べるため、収束性能を評価する指標として平均評価回数と、大局的探索性能を評価する指標として成功率を用い、また、目的関数としては、単峰性か多峰性か、対称か非対称か、変数分離可能か不可能かの分類の出来る 9 関数を用意した。PfGA は、SSGA と比較しても収束の速いアルゴリズムであり、また、9 関数全てに対しても探索に成功する確率が SSGA よりも良いことが確認され、PfGA の優れた探索能力が示された。

7.1.2 並列分散処理への拡張

第3章では、探索における解への収束速度を向上させる目的で、PfGA の並列分散処理について述べた。並列処理がその効果を発揮できるのは、特に探索問題の場合であるからである。また、少数個体をベースとした PfGA では、複数の局所集団を並列化することにより並列処理の実装が容易であり、同時に、GA における集団の多様性というアプローチも、この並列分散処理によって容易となった。アルゴリズムは、並列化された局所集団毎の役割で 2 種類、すなわち一様分散型とマスタースレーブ型に分類し、また、局所集団間での個体の移民の仕方で 2 種類に分類し、合計 4 種類の実装法によって、並列化の実験を行った。目的関数としては前述の 9 関数を用いた。その結果、一様分散型で、直接移民法が比較的良好な結果を得ることが分かった。また部分的に、成功率を向上させる結果も得ることができ、並列処理における移民の効果を確認することが出来た。

7.1.3 遺伝子重複による拡張

第4章では、成功率を向上させる目的で、遺伝子重複 GA について述べた。遺伝子重複を新たな遺伝的オペレータととらえ、GA の中でこれを効果的に働かせる機構として、遺伝子

の連結, 伸長, 結合などの4種類の重複方法を提案した. ベースとなるアルゴリズムとしては PfGA, および SSGA を用い, ベンチマーク問題として9関数を解かせることにより, 遺伝子重複の効果を検証した. 実験の結果, ほぼ全ての関数に対して重複の効果が現れ, ベースとなる GA に無関係に最良の重複方法が関数ごとに存在することが分かった. また, ベースとなる GA は SSGA よりも PfGA の方が優れているが, 多峰性, 非対称, 変数分離不可能を同時に持つような関数に対しては, PfGA ベース, SSGA ベース共に, 重複による成功率の向上は現れなかったが, SSGA ベースの場合は収束性能の向上が見られた.

7.1.4 統計力学的計算法と進化的計算法との比較

第5章では, 進化的計算法以外の探索アルゴリズムである, 統計力学的計算法との対比を示した. スピン系におけるエネルギーの最小値を求めるための手法として知られるモンテカルロ法 (MC) や量子モンテカルロ法 (QMC) に, シミュレーテッドアニーリング法 (SA) を併用する計算手法が, 実は GA と深い関わりがあり, GA の一部としてとらえることが可能である. そこで, MC(QMC) と PfGA との比較を行った結果, MC, QMC は PfGA に比べ収束性能に優れていることが確認されたが, 多峰性関数に対する成功率は MC, QMC 共に PfGA に比べ小さく, 大域的な探索能力は低いということも確認された.

また, GA に対する理論解析が困難な理由として, 個体数が可変であることを挙げた. 個体数が可変であるような系は, 物理では粒子数が可変な非平衡系であるため, 非平衡系における分布関数の概念を考慮すれば, 様々なモデルに対するシミュレーションが可能となることを述べた. さらに, この非平衡系の計算モデルを援用することにより, PfGA の変則的な選択方法を理論的に解析する可能性を示した.

7.1.5 理論解析

第6章では, 理論解析の一助として, 比較的, 解析が容易である突然変異による効果についての解析を行った. 突然変異によって個体がどの程度移動するかを, 定量的な解析と実験により検証した. その結果, 摂動を受けた個体の分布は裾野が広がった分布を示し, 突然変異率に比例した分散を持つことが分かった. 交叉については, 親個体の座標を固定した場合の, 子個体の発生場所を統計的に求めた. その結果, 親2個体のハミング距離が大きいときは, 生成される子個体は全探索空間に広く分布し, 逆にハミング距離が小さいときは, 子個体は親個体に近い場所に生成されることが確認された. これにより, GA において交叉が本質的に重要であることの理由が明らかとなった. また, 本研究で用いた2つの評価基準 (成功率と平均評価回数) に対する考察を行い, これらがある分布関数を介して密接に関係していることを示した. この分布関数の積分値が成功率を表しているため, 並列処理による探索の改善などの議論がより明確となった.

7.2 結論と今後の課題

本論文では、優れた探索アルゴリズムを開発する目的で、進化的計算法のうち、特に遺伝的アルゴリズムとその拡張についての研究と、様々な最適化問題への適用に関する研究を述べた。優れた確率的探索アルゴリズムとは、汎用的で設計が容易なアルゴリズム、収束性能が優れたアルゴリズム、および探索における成功率が高いアルゴリズムに分類されると考え、これらの目的に対するアルゴリズムをそれぞれ提案し、性能評価を行った。また、他の探索アルゴリズムであるモンテカルロ法が、アルゴリズムの構造上、遺伝的アルゴリズムに含まれるものであることを見出し、これについても性能評価を行った。

汎用的で、かつ設計が容易であるようなアルゴリズムは互いに矛盾した目標である。すなわち、どんな問題も解くことができる代わりに多くのパラメータを適切に設計しなければならない。パラメータフリー遺伝的アルゴリズムはこの矛盾を見事に解決したと言える。また、収束性能を向上させることと、成功率を向上させることも互いに相反する目標であるが、並列分散処理や、遺伝子重複を実装した遺伝的アルゴリズムは、部分的にはあるが、収束性能と成功率の両方を向上させることができた。

モンテカルロ法のような統計力学的計算法は、遺伝的アルゴリズムに比べ、比較的局所探索が得意であるが、これを大域的探索に拡張するような方法はおそらく、非平衡系モデルをベースとするか、あるいは遺伝的アルゴリズムと組み合わせたアルゴリズムを構築しなければならないであろう。これに対し、遺伝的アルゴリズム自体はまさに、非平衡多体系である生物の進化や生態系から着想を得た計算法である。最適化の道具としては、統計力学的計算法の方が歴史が古いにもかかわらず、ベースとなるモデルに関しては、進化的計算法の方が奥が深いとも言えるのである。

今後の課題としては、以下のものが挙げられる。

- 遺伝的アルゴリズムの目的関数に応じた理論解析
理論解析によって、より最適な進化機構が得られる可能性があり、同時に、生態系の理論モデルにも援用することができると考えられる。
- 組み合わせ最適化問題に対する解法
組み合わせ最適化問題に関しては本論文では取り上げなかったが、これを遺伝的アルゴリズムで解く場合、codingの問題が特に重要である。よって、なんらかのcoding方法や、またはcodingによらない計算方法などの発想が必要であろう。

謝辞

本論文を作成するにあたり、総務省通信総合研究所けいはんな情報通信融合研究センターの澤井秀文総括主任研究官(神戸大学大学院システム機能科学専攻併任教授)には指導教官・主査として御指導、御鞭撻頂きましたことを深く感謝致します。また、本論文を洗練するために貴重な御助言を頂いた神戸大学大学院システム機能科学専攻長の阿部重夫教授、ならびに同大学院情報メディア科学専攻の北村新三教授に深く感謝申し上げます。

本研究のうち、第2章で取り扱ったパラメータフリー遺伝的アルゴリズムについては、東芝 iValue Creation Company の木津左千夫氏のアイデアによるものであり、JAVA プログラムを提供して頂きました。同氏に深く感謝致します。

最後に、総務省通信総合研究所関西支所の福地一支所長をはじめ、同研究所知覚機構研究室の皆様には、潤沢な環境と有益な御指導を与えて頂きましたことを心より感謝申し上げます。

参考文献

- [1] Teukolsky, Vetterling, and Flannery, “Numerical Recipes in C,” Cambridge University Press (1993).
- [2] Charles Darwin, “On the Origin of Species by Means of Natural Selection or the Preservation of Favoured Races in the Struggle for Life,” London, John Murray (1859).
- [3] J. H. Holland, “Adaptation in Natural and Artificial System,” The University of Michigan Press (1975).
- [4] De Jong, “An Analysis of the Behavior: A Class of Genetic Adaptive Systems,” Doctoral dissertation, University of Michigan (1975).
- [5] D. E. Goldberg, “Genetic Algorithms in Search, Optimization, and Machine Learning,” Addison Wesley (1989).
- [6] T. Baeck, D. Fogel and Z. Michalewicz (Eds.), “Handbook of evolutionary computation,” New York: Oxford University Press (1997).
- [7] G. Syswerda, “A Study of Reproduction in Generational and Steady-State Genetic Algorithms,” Foundation of Genetic Algorithms, pp. 94-101, Morgan Kaufmann (1991).
- [8] 北野宏明編, “遺伝的アルゴリズム,” 産業図書 (1993)
- [9] 北野宏明編, “遺伝的アルゴリズム 3,” 産業図書 (1997)
- [10] 北野宏明編, “遺伝的アルゴリズム 4,” 産業図書 (2000)
- [11] 伊庭斉志, “進化論的計算の方法,” 東京大学出版会 (1999)
- [12] 木津, 澤井, 足立, “可変な局所集団の適応的探索を用いたパラメータフリー遺伝的アルゴリズムとその並列分散処理への拡張,” 電子情報通信学会論文誌, Vol.J82-D-II, No.3, pp. 512-521 (1999).
- [13] S. Tsutsui, Fujimoto and A. Ghosh, “Forking Genetic Algorithms: GAs with Search Space Division Schemes,” Evolutionary Computation, pp. 61-80, vol. 5, no. 1, MIT Press (1997).

- [14] F. Vavak, K. Jukes and T. C. Fogarty, "Adaptive Combustion Balancing in Multiple Burner Boiler Using a Genetic Algorithm with Variable Range of Local Search," Proceedings of the Seventh International Conference on Genetic Algorithms, pp. 719-726 (1997).
- [15] J. M. Varanelli and J. P. Cohoon, "Population-Oriented Simulated Annealing: A Genetic/Thermodynamic Hybrid Approach to Optimization," Proceedings of the Sixth International Conference on Genetic Algorithms, pp. 174-181, Morgan Kaufmann (1995).
- [16] The Organising Committee: H. Bersini, M. Dorigo, S. Langerman, G. Seront, L. Gambardella, "Results of the First International Contest on Evolutionary Optimization (1st ICEO)," 1996 IEEE International Conference on Evolutionary Computation (ICEC'96), pp. 611-615 (1996).
- [17] G. Bilchev and I. Parmee, "Inductive Search," 1996 IEEE International Conference on Evolutionary Computation (ICEC'96), pp. 832-836 (1996).
- [18] D-G. Li and C. Smith, "A New Global Optimization Algorithm Based on Latin Square Theory," 1996 IEEE International Conference on Evolutionary Computation (ICEC'96), pp. 628-630 (1996).
- [19] R. Storn and K. Price, "Minimizing the Real Functions of the ICEC'96 Contest by Differential Evolution," 1996 IEEE International Conference on Evolutionary Computation (ICEC'96), pp. 842-844 (1996).
- [20] C. H. M. van Kemenade, "Cluster Evolution Strategies - Enhancing the Sampling Density Function Using Representatives -," 1996 IEEE International Conference on Evolutionary Computation (ICEC'96), pp. 637-642 (1996).
- [21] G. Seront and H. Bersini, "Simplex GA and Hybrid Methods," 1996 IEEE International Conference on Evolutionary Computation (ICEC'96), pp. 845-848 (1996).
- [22] M. Furusawa and H. Doi, "Promotion of Evolution: Disparity in the Frequency of Strand-specific Misreading between the Lagging and Leading DNA Strands Enhances Disproportionate Accumulation of Mutations," J. theor. Biol., vol. 157, pp. 127-133 (1992).
- [23] 足立進, 澤井秀文, "並列分散パラメータフリー遺伝的アルゴリズムにおける移民選択法の効果," 電子情報通信学会論文誌, Vol. J83-D-I, No.8, pp. 834-843 (2000).
- [24] R. Tanese, "Distributed Genetic Algorithms," Proceedings of the Third International Conference on Genetic Algorithms, pp. 434-439, Morgan Kaufmann (1989).

- [25] B. Manderick and P. Spiessens, "Fine-Grained Parallel Genetic Algorithms," Proceedings of the Third International Conference on Genetic Algorithms, pp. 428-433, Morgan Kaufmann (1989).
- [26] P. Spiessens and B. Manderick, "A Massively Parallel Genetic Algorithm Implementation and First Analysis," Proc. of the Fourth Int. Conf. on Genetic Algorithms, pp.279-286, Morgan Kaufmann, 1991.
- [27] H. Muehlenbein, M. Schomisch and J. Born, "The Parallel Genetic Algorithm as Function Optimizer," Proceedings of the Fourth International Conference on Genetic Algorithms, pp. 271-278, Morgan Kaufmann (1991).
- [28] E. C-Paz and D. E. Goldberg, "Predicting Speedups of Idealized Bounding Cases of Parallel Genetic Algorithms," Proceedings of the Seventh International Conference on Genetic Algorithms, pp. 113-126, Morgan Kaufmann (1997).
- [29] T. Maruyama, T. Hirose and A. Konagaya, "A Fine-Grained Parallel Genetic Algorithm for Distributed Parallel System," Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 184-190, Morgan Kaufmann (1993).
- [30] Al Geist, Adam Beguelin, Jack Dongarra, Weicheng Jiang, Robert Manchek and Vaidy Sunderam, "PVM UER'S GUIDE AND REFERENCE MANUAL," (1993).
- [31] Hideo Sawai and Susumu Adachi, "Genetic Algorithm Inspired by Gene Duplication," Proceedings of the 1999 Congress on Evolutionary Computation (CEC99), vol.1, pp. 480-487, Washington D.C., July (1999).
- [32] Susumu Ohno, "Evolution by Gene Duplication," Springer-Verlag (1970).
- [33] 金子邦彦, 池上高志, "複雑系の進化的シナリオ," 複雑系双書 2 (1998).
- [34] Lent, C. S. Tougaw, P. D. Porod, and W. Bernstein, Nanotechnology, Vol.4, No.49 (1993).
- [35] Susumu Adachi and Yoshimasa Isawa, "Cell Design and Dynamics of Quantum Cellular Automata," Solid-State Electronics, Vol. 42, No. 7-8, pp. 1361-1366 (1998).
- [36] K. Binder, "Monte Carlo Methods in Statistical Physics," Springer, Berlin (1979).
- [37] A. Wiesler, Phys. Lett. 89A, pp. 359 (1982).
- [38] M. F. Herman, E. J. Bruskin and B. J. Berne, J. Chem. Phys. 76, pp. 5150 (1982).
- [39] M. Suzuki, Prog. Theor. Phys. 56, pp. 1454 (1976).
- [40] T. Onogi, S. Miyashita, and M. Suzuki, "Quantum Monte Carlo Methods," Springer-Verlag, Berlin p. 75 (1987).

- [41] S. Adachi, K. Fujimoto, T. Hatano and Y. Isawa, "Current Response of Quantum Dot Modulated by Time-Dependent External Fields," *Jpn. J. Appl. Phys.* 34 p. 4298 (1995).
- [42] 伊澤義雅, 羽田野剛, 足立進 "半導体量子ドットにおけるクーロン・ブロッケード" 電子情報通信学会, 信学技報, ED96-219, pp. 25 (1997.3).
- [43] B. A. Berg and T. Neuhaus, "Multicanonical Ensemble: A New Approach to Simulate First-order Phase Transitions," *Physical Review Letters*, 68 pp. 9-12 (1992).
- [44] Per Bak, "Self-Organized Criticality: A Holistic View of Nature," *Complexity: Metaphors, Models, and Reality*, G. Cowan, D. Pines, and D. Meltzer, (Eds.) SFI Studies in the Science of Complexity, Proc. Vol. XIX, Addison-Wesley (1994).