Kobe University Repository : Kernel

PDF issue: 2025-05-11

# 進化戦略における自己適応の拡張に関する研究

松村，嘉之

(Degree)
博士（工学）
(Date of Degree)
2002-09-30
(Date of Publication)
2008-11-12
(Resource Type)
doctoral thesis
(Report Number)
甲2662
(URL)
https://hdl.handle.net/20.500.14094/D1002662

※ 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。

博 士 論 文

# 進化戦略における自己適応の拡張に関する研究

平成14年8月

神戸大学大学院自然科学研究科

松 村 嘉 之

# 目 次

# 第1章　緒論

## 1.1　研究の背景

現実のシステムは様々に競合する部分問題が無数かつ複雑に絡み合っており，分割して分析することが困難な複雑システムとして捉えることができる．例えば，この複雑システムを対象とした最適化問題を扱う場合，多峰性であったり，ノイズを含んでいたり，不連続であったりするため，最適性を保証する限定的に有効な最適化手法では最適化が実質的に困難となることが多い．そのため，複雑なシステムを対象とする場合は，最適性の保証がなくとも実用的な実行可能解を求めるための系統的な計算手法が必要とされている．そこで，進化型計算 (Evolutionary Computation : EC) [10][14][84] は，高次元の非線形問題やノイズを含む問題等に対してもより良い近似解を求めることが可能である点と，幅広い問題に対し頑健である点で特に有望視されている．

この EC には，1960 年代に提唱された進化戦略（Evolution Strategies : ES)[13][104]，進化的プログラミング（Evolutionary Programming : EP)[51]，遺伝的アルゴリズム（Genetic Algorithms : GA）[56][66] の 3 つの主な領域があり，目的変数の表現形式，突然変異，組換え，選択等により特徴付けられる．特に目的変数を実数値表現とする場合，ES は初期から実数値表現に着目しているのに対し，EP と GA は 1990 年代に実数値表現の研究が盛んに行われるようになった [49][51][91][92][100][110][114]．これらの研究の中でも，実数値表現の目的変数と変化量を制御する戦略パラメータの 2 変数を個体表現とし，進化過程で目的変数と戦略パラメータを進化させる手法は，優れた性能を示している [10][24][41][71][98][99][101]．

しかしながら，高次元の非線形問題や超多峰性問題を取り扱った場合，適応能力は不十分であるため，戦略パラメータが大域最適解に到達する以前に非常に小さくなる傾向が現れる．その結果，突然変異のステップサイズも極めて小さくなり，実質的に探索点が動かない状態に陥り集団が収束する，いわゆる早期収束が起こりやすくなる．一般には，これを回避するために，戦略パラメータに下限を設定して小さくなり過ぎないようにする．このことに関して，K.-H. Liang らは問題特性に依存した最適な下限を戦略パラメータに設定しなければならないと報告している [73]．この問題特性にあわせた最適な下限を調節することは，(1) それ以上ステップサイズが小さくならず局所探索の能力を衰えさせてしまうことや，(2) 問題の特性に合わせた最適な下限を調節することは本来進化型計算が目指している頑健性の観点からは好ましくないことから，望ましい解決法とは言い難い．そこで，彼らは下限を動的に制御する手法を試みてい

るが，これも頑健性の観点で不十分な結果となっている [74][75].

今後，ECによってさらに複雑なシステムを取り扱う場合，高次元の非線形問題や超多峰性問題であっても，早期収束の状態を回避できる頑健性を備えた手法を構築することが必須と考える.

## 1.2 研究の目的

本論文はECによって複雑なシステムを取り扱うことを目指し，高次元の非線形問題や超多峰性問題であっても，早期収束の状態を回避できる頑健な進化型計算手法を構築することを目的とする．そこで，上記の早期収束の現象を回避するため，適応能力を向上させる手法を新たに構築する．この手法の基本的アイデアは，戦略パラメータに対し選択圧によらない確率的変化要因を加え，特定の値に固着しにくくしようとするところにある．それゆえ，各有効な戦略パラメータの他に冗長な戦略パラメータを持つ個体表現法を採用し，有効なものと冗長なものとを確率的に入れ替える新しい突然変異手法を提案する．そして，標準テスト関数に対する基本性能およびノイズに対する性能を計測し，提案手法と代表的な従来手法との比較を行うことにより，提案手法の有効性を実証する．進化ダイナミクスの解析を行うことにより手法の特徴を明らかにする．また，工学的実問題への応用を通して手法の有用性を検証する．さらに，組換え操作による自己適応の補完に注目し，組換え操作と自己適応の相性を検証する.

## 1.3 本論文の構成

第1章では，本論文の背景と目的，及び論文全体の構成を示している.

第2章では，本論文が対象とする進化型計算手法のうち1960年代に始まった進化戦略，進化的プログラミング，遺伝的アルゴリズムの3つの手法の概略を紹介する．次に，本論文が取り扱う実関数最適化問題を定義し，この問題領域において優れた性能を示す自己適応について説明する．特に，進化戦略，進化的プログラミング，遺伝的アルゴリズムにおける自己適応の関連研究を示して研究領域を明らかにし，さらに，自己適応の問題点を指摘する.

第3章では，本論文において提案する手法が動機付けられた生物進化メタファについて述べ，戦略パラメータの自己適応性を向上させる手法の基本的アイデアを説明する．そして，各有効な戦略パラメータの他に冗長な戦略パラメータを持つ個体表現法を示し，冗長なものと有効なものを確率的に入れ替える新しい突然変異手法を定式化する．次に，自己適応を補完するMulti-parent Recombination(MPR)について関連研究を述べ，本論文が参照する3つのMPRについて説明する.

第4章では，進化的プログラミングにおける自己適応の拡張を取り扱う．実関数最適化問題における標準テスト関数を用いた計算機実験により，従来の進化的プログラ

ミング (Classical-EP: CEP, Fast-EP: FEP) は，往々にして戦略パラメータが極端に小さくなり，集団が実質的に動かない状態になりやすいことを明らかにする．それに対し，冗長な個体表現法と適切な突然変異パターンによって遺伝的浮動効果を戦略パラメータの変化要素に加える手法 (Robust-EP: REP) を提案する．そして，標準テスト関数を用いた計算機実験によりその有効性を検証し，進化ダイナミクスを観測することで提案手法の特性を明らかにする．また，ノイズを含むテスト関数に関してもその有効性を検証する．さらに，工学的実問題への応用として，多指ハンドによる物体の把持問題に適用し，非線形制約条件つき最適化問題に対する有用性を検証する．

第5章では，進化戦略における自己適応の拡張を取り扱う．実関数最適化問題における標準テスト関数を用いた計算機実験により，従来の進化戦略 (Classical-ES: CES, Fast-ES: FES) は，進化的プログラミングの場合と同様に，往々にして戦略パラメータが極端に小さくなり，集団が実質的に動かない状態になりやすいことを明らかにする．それに対し，冗長な個体表現法と適切な突然変異パターンによって遺伝的浮動効果を戦略パラメータの変化要素に加える手法 (Robust-ES: RES) を提案する．そして，標準テスト関数を用いた計算機実験によりその有効性を検証し，手法の特性を知るために突然変異率と問題の次元スケールによる進化ダイナミクスの変化を観測する．また，ノイズを含むテスト関数に関してもその有効性を検証する．さらに，工学的実問題への応用として，自律移動ロボットのナビゲーション問題における Continuous-Time Recurrent Neural Networks の進化的設計を取り扱い，従来手法 (CES, FES) と拡張手法 (RES) の性能を検証する．

第6章では，進化戦略における MPR による自己適応の補完を取り扱う．従来手法 (CES, FES) と提案手法 (RES) に関して，Intermediate Recombination と Discrete Recombination を実数値変数と戦略パラメータの両方に用いた場合における MPR の性能を検証し，その進化ダイナミクスを解析する．また，ノイズを含むテスト関数に関してもその有効性を検証する．さらに，工学的実問題の応用として，Evolutionary Artificial Neural Networks の構造進化を取り上げ，MPR の性能検証を行う．

第7章では，本論文の結論を述べる．

# 第2章　進化型計算手法の現状，自己適応の問題点

## 2.1　緒言

本章では，本論文において対象とする進化型計算手法のうち1960年代に始まった進化戦略，進化的プログラミング，遺伝的アルゴリズムの3つの手法の概略を紹介する．次に，本論文が取り扱う実関数最適化問題を定義し，この問題領域における自己適応を説明する．特に，各手法に関する自己適応の関連研究について述べ，研究領域を明らかにし，自己適応の問題点を指摘する．

## 2.2　進化型計算手法の概略

Bremermann，Friedman，Box等，1950年代や1960年代の計算機科学者達は，最適化問題を解くための道具として進化を捉えることができると考え，進化システムの研究を始めた．特に，1960年代にはRechenbergとSchwefelが進化戦略を提唱し，実数値変数最適化に用いた [13][104]．そして，Finite State Machineを進化させるアルゴリズムとしてFogelやOwensやWalshらによって進化的プログラミングが提唱された [51]．また，Hollandによって遺伝的アルゴリズムが提唱された [66]．特に，進化戦略や進化プログラミングと対比すると，Hollandの元々の目的は，特別な問題を解くためのアルゴリズムを作成することではなく，むしろ自然のような適応的現象を定式化することであり，自然淘汰のメカニズムを計算機上で表現するための方法を発見することであったように思われる．

近年では，1960年代に始まった進化戦略，進化的プログラミング，遺伝的アルゴリズムの3つの研究領域ならびに関連研究領域の研究者が集まり，進化型計算と呼ばれる分野を形成している．現在では研究分野が広がり，研究者の相互関係が深まるにつれて，進化戦略，進化的プログラミング，遺伝的アルゴリズム等の進化的なアプローチの境界が曖昧になりつつあるが，以降，これらの一般的な枠組みについて述べる．

### 2.2.1 進化戦略

進化戦略 (Evolution Strategies:ES) [104][13] は，1964 年にドイツの Technical University of Berlin で Flashing Nozzle の設計に際して開発されたのが始まりである．翌年には，Rechenberg と Schwefel によって，1 個体の親が 1 個体の子孫を生む $(1+1)$-ES [4] が発表された．これは，two membered ES と呼ばれ，突然変異と選択の単純なものであった．その後，multimembered ES として $\mu$ 個体 ($\mu$ >1) の親から 1 個体の子孫を生成し，最も劣った親と置き換える $(\mu+1)$-ES を経て，さらに，1 個体の親から $\lambda$ 個体の子孫を生成する $(1,\lambda)$-ES [21][25][27][31] を経て，並列計算と自己適応を可能とする目的のために，1977 年頃には $\lambda$ 個体の子孫を生成する $(\mu,\lambda)$-ES ($\mu \leq \lambda$) [22][2] に拡張された [8][105]．この $(\mu,\lambda)$-ES は $\mu(>1)$ 個の個体からなる親集団から，突然変異を経て $\lambda$ 個 ($\lambda > \mu$) の子孫を生成し，適応度の高い順に $\mu$ 個を親として選び次世代集団を形成する．なお，エリート戦略を用いる $(1+\lambda)$-ES と $(\mu+\lambda)$-ES は探索能力が弱まる [8][13][93] と言われており，あまり用いられない．さらに，近年，Beyer [23] によって $\rho$ 個体の組換え操作を用いる $(\mu/\rho,\lambda)$-ES ($1 < \rho \leq \mu$) [1][3][94] が解析され，特に，$\rho$ >2 の場合は multirecombination と呼ばれている [29].

(1+1)-ES において，Kappler [68] は，突然変異の計算で Gauss 分布の代わりに Cauchy 分布を用いて，$(1+1)$-ES のパフォーマンスを計測している．彼は，低次元の探索問題に対して有効性を見出したものの，高次元問題に対しては明確な有効性は分からないとしていた．しかし，これに触発された Yao と Liu [117][119] は，$(\mu,\lambda)$-ES において，突然変異の計算で Gauss 分布の代わりに以下に示す式で $t=1$ とした Cauchy 分布関数 $F_t(x)$ に基づく乱数値を用いることを提案した：

$$F_t(x) = 1/2 + (1/\pi)\tan^{-1}(x/t) \tag{2.1}$$

彼らは数多くのテスト関数を用いて計算機実験を行い，これが多くの問題，特に多峰性問題に有効であることを示し，Gauss 分布を用いた Classical ES (CES) と区別して，Fast ES (FES) と呼んだ．

$(\mu/\rho,\lambda)$-ES は，$\mu$ 個の親個体から組換えや突然変異を経て $\lambda$ 個の子孫を生成し，子孫の中から適応度の高い順に $\mu$ 個を親個体として選択し，次世代集団を形成する．ここで，$\rho$ は組換え操作において 1 個体の子孫を生成するのに必要な親個体の個数を表している．この $(\mu/\rho,\lambda)$-ES に関して，Beyer [23] は $\rho$=$\mu$ の場合の $\mu$-fold speedup を理論的に証明している．ES の個体は実数値変数と戦略パラメータの 2 変数を持つが，この証明において戦略パラメータに組換え操作を用いていない．実数値変数の突然変異の大きさを戦略パラメータが大まかに規定することを考えると，この証明では不十分である．そのため，実数値変数だけでなく戦略パラメータに対しても組換え操作を用いた計算機実験や理論的考察によって解析を行わなければならない．

進化型計算手法における ES のアルゴリズム上の特徴として，(1) 二進数等にコード化された遺伝子型を用いずに実数を直接並べた個体を取り扱う，(2)2 個体による組換

えよりも1個体上の突然変異を主たるオペレータとする，(3) 確定的選択過程を用いる，等が挙げられる．

なお，次節で述べる進化的プログラミングは確率的選択を用いる $(\mu+\mu)$-ES の一種とみなすことができる．

ES アルゴリズムの流れを以下に示す．

1. ランダムに $\mu$ 個の個体からなる初期集団を発生させ，$gen = 1$ とする．
2. 親個体の適応度を目的関数 $f(\boldsymbol{x}_i)$ を元に計算する．
3. $\mu$ 個体の親から組換えと突然変異によって $\lambda$ 個体の子孫を生成する．
4. 子孫の個体の適応度を $f(\tilde{\boldsymbol{x}}_i^k)$ を元に計算する．
5. $\lambda$ 個の子孫を適応度の高い順に並べ，良いものから $\mu$ 個を次世代の親として選択する．
6. 終了条件を満たさない時は $gen = gen + 1$ として 3 に戻る．

### 2.2.2 進化的プログラミング

進化的プログラミング (Evolutionary Programming: EP) は，1960 年代前半に L. J. Fogel が有限状態機械を進化させる枠組みとして提案したものを，その息子 D. B. Fogel が 1990 年代初頭に "meta-EP" として実関数最適化問題向けに定式化し直したものである [51]．なお慣習に習い，以下ではこれも EP と呼ぶ．EP の計算手法自体は，D. B. Fogel による定式化以来よく知られており，多くのベンチマーク問題だけでなく実問題 (例えば，[53]) でもその優れた探索能力が検証されている．しかしながら，EP 自身の最適化能力の向上を図った研究はそれほど多くはない．これは，EP が頑健で十分な探索能力を持つためでなく，どの計算過程に拡張を加えれば良いのか絞りきれなかったためと推察する．そのため，主として再生過程における突然変異のための確率分布様式に焦点を当てた例がいくつかある．

Yao と Liu [116][121] は EP の拡張手法の一つとして，Gauss 型突然変異の代わりに Cauchy 型突然変異を用いる手法を提案した．これは，Gauss 分布の代わりに，以下の Cauchy 分布の確率密度関数 $f(x)$ において $\alpha = 1$，$\beta = 0$ として得られる乱数値を用いるものである：

$$f(x) = \alpha/\pi(\alpha^2 + (x - \beta)^2) \tag{2.2}$$

彼らは，従来の Gauss 分布を用いる手法を Classical Evolutionary Programming (CEP) と呼び，Cauchy 型突然変異を用いる手法を Fast Evolutionary Programming (FEP) と呼んで区別した．特に，この FEP は多峰性の問題に対し有効であることが示されてい

る．そして，このFEPのパフォーマンスの向上は戦略パラメータが0に近づいた時にもGauss型突然変異よりもCauchy型突然変異の方が比較的大きな変異が生成されるためと分析している[118][121].

この報告に触発され，Gauss分布とCauchy分布の確率密度関数を平均した分布[36][37][38][39]やLevyの確率分布[72]等，他の確率分布に関する議論や，Gauss分布とCauchy分布の両方を同時に用いる手法[102][120][121]等，確率分布に関して様々な議論が展開されている．

なお，EPは，ESとほぼ同じ手続きによって構成されるが，(1)種の進化をモデル化しているため，組換えのための演算は適用されない，(2)確率的選択過程を用いる，等の特徴があげられる[11].

EPのアルゴリズムを以下に示す.

1. 世代$g=0$とし，$\mu$個体からなる初期集団をランダムに発生させる.
2. 親個体の適応度を目的関数$f(\boldsymbol{x}_i)$を元に計算する.
3. $\mu$個体の親から$\mu$個体の子孫をそれぞれ再生する.
4. 子孫の個体の適応度を$f(\boldsymbol{x}_i)$を元に計算する.
5. $\mu$個体の親と$\mu$個体の子孫の合計$2\mu$個体からランダムに選ばれた$q$個体の適応度を比較して各個体の勝ち点を計算する．$2\mu$個体の集団から勝ち点の高い$\mu$個体を次世代の親として選択する.
6. 終了条件を満たさない時は$g=g+1$としステップ3に戻る.

### 2.2.3 遺伝的アルゴリズム

遺伝的アルゴリズム (Genetic Algorithms: GA)は，生物の進化過程をモデル化したアルゴリズムであり，組合せ最適化問題等幅広い問題に適用されている．GAで使われている概念の多くは，Holland [66]の"Adaptation in Natural and Artificial Systems"に記述されている.

GAは生物進化の遺伝メカニズムを取り入れた学習方法であり，これは離散システムの最適化のみならず，様々な最適化問題に応用可能な手法である．実際に最も多いGAの応用は最適化に関するものであるが，Holland自身はGAを適応アルゴリズムであると定義づけている．また，Hollandらの研究にCognitive System 1 (CS1)という学習システムの構成があり，GAを最適化手法というよりも学習システムと考えていたようである．このCS1は分類子システム (Classifier System)と呼ばれている.

最適化は，Hollandに学んだDe Jongの研究を契機として，大きく発展した．そして，GrefenstetteがGENESISというGAのフリーウエアを作成し，GA研究の普及をはかっ

た．しかし，GA研究の普及に最も大きな影響力を与えたのは，1989年のGoldbergの著書"Genetic Algorithms in Search, Optimization and Machine Learning" [56] であり，この本には理論とともに数多くの例題が解説されている．その後のGAの拡張として，不定長な遺伝子表現の木構造を用い計算機のプログラミングを進化させる遺伝的プログラミング(Genetic Programming)が1992年にKozaによって提案されている．

実数値表現を直接用いる関数最適化手法はReal-coded GAと呼ばれ，1990年代に入って研究が盛んに行われるようになり [110][114]，特に，交叉に焦点を当てた研究は盛んに行われている [49][91][92]．

GAのアルゴリズムの流れを以下に示す．

1. ランダムに$\mu$個の個体からなる初期集団を発生させ，$gen = 1$とする．
2. 集団中の各個体の適応度を目的関数$f(\boldsymbol{x}_i)$を元に計算する．
3. 各個体について，その適応度に応じて次世代に残す子孫(offspring)の数を増減させる選択を行う．
4. 集団内の個体をランダムにペアリングし，ある確率(交叉率：crossover rate)で遺伝子を部分的に交換する交叉を行う．
5. 各個体について，ある確率(突然変異率：mutation rate)で，各遺伝子座の遺伝子を対立遺伝子と入れ換える突然変異を行う．
6. 終了条件を満たさない時は$gen = gen + 1$として3に戻る．

## 2.3 研究領域

以下に進化型計算を様々な問題に応用した例を示す．

1. 最適化： 関数最適化や，回路設計やスケジューリング等の組合せ最適化問題に広く使われている．
2. 自動プログラミング： 計算機プログラミングの評価や，セルラーオートマタや，ソーティングネットワークのような計算機構造を設計するために使われている．
3. 機械学習： タンパク質構造や，天気予報のような予想や分類を含む多くの機械学習に使われている．さらに，ロボット制御のためのシンボル生成システムや，学習分類システム等のルール作り，人工神経回路網の設計等，特定の機械学習を評価するために使われている．
4. 経済学： ゲーム理論や，入札戦略の開発，経済市場の形成等の過程をモデル化するために使われている．

5. 生態系： 種の進化，共生，共進化，資源循環のような生態現象のモデル化をするために使われている．

6. 集団遺伝学： 進化的な集団を再形成するために，遺伝子をどのような状態にすればよいのかという集団遺伝学の研究に使われている．

7. 社会システム： 社会性昆虫の集団としての振る舞いや，マルチエージェントシステムにおける通信や協力の進化過程等，社会システムに関する研究に使われている．

これら以外にも様々な問題に進化型計算は適用されているが，本論文は関数最適化において個体表現に実数値を直接用いる実関数最適化に焦点を当てている．

### 2.3.1 実関数最適化問題

本論文が対象とする実関数最適化問題とは，$n$ 次元の実行可能領域 $S(S \subseteq R^n)$ を持った有界な実数値目的関数 $f(\boldsymbol{x})$ において，$f(\boldsymbol{x}_0)$ が最小値となる決定変数ベクトル $\boldsymbol{x}_0 \in S$ を見つける問題である．この時，実関数最適化問題は以下のように定式化される．

$$\boldsymbol{x}_0 = \arg \min_{\boldsymbol{x} \in S} f(\boldsymbol{x}) \tag{2.3}$$

なお，最大値を求める問題は負の記号を用いることによって，一般性を失うこと無く最小値を求める問題として扱えるため，本論文では最小値を求めるテスト問題を考える．

また，本論文で取り扱う問題は，ある探索空間内の一点 $\boldsymbol{x}$ とその近傍の点 $\boldsymbol{x}'$ では，$f(\boldsymbol{x})$ と $f(\boldsymbol{x}')$ の間に，ある程度の相関があるとする：

$$\boldsymbol{x} \simeq \boldsymbol{x}' \Longrightarrow f(\boldsymbol{x}) \simeq f(\boldsymbol{x}') \tag{2.4}$$

すなわち，緩やかな丘が連なる landscape に大域最適値をとる急進なスパイクが刺さっているような問題や，完全にランダムな関数値をとる問題等を取り除いた問題範囲を扱うものとする．この定義は曖昧であるものの，アルゴリズムの探索能力に関する比較が意味をなすことが実質的に可能である限定された問題範囲を指し示す．すなわち，NFL 定理 (No-Free-Lunch Theorems) [40][111][112] が成立する一般問題の範囲を取り扱わないことを意味し，工学的問題によくある landscape 特徴であると思われる式 (2.4) が成り立つ関数のみを取り扱うこととする．このような問題の範囲では，最適化アルゴリズムの評価が可能であると考える．

進化型計算 (Fig.2.1) には 1960 年代に始まった進化戦略 (ES)，進化的プログラミング (EP)，遺伝的アルゴリズム (GA) の 3 つの主な研究領域があり，GA はその創成時

期には個体表現に主にバイナリ表現を用い，ESは実数値を直接個体表現として実関数最適化に用いる．EPはその創成時期には有限状態機械の入出力記号を個体表現として用いていたが，近年のFogelのmeta-EP以降，実数値を直接個体表現として用いる実関数最適化の研究が多く行われるようになった．また，GAでは1990年代以降個体表現に直接実数値を用いる実関数最適化の研究が盛んに行われるようになった．

本論文は，実数値を直接個体表現として用いる進化型計算の中でも，特に優れた性能を示しているEPとESの自己適応に着目して研究を進める．なお，EPとESの違いは，選択に関して確率的か，確定的かという点にあり，EPは組換え操作を用いないという特徴を持つ[45]．本論文ではEPを第4章で扱い，組換え操作を用いないESを第5章で，そして，Beyer [23]が解析しこれに動機付けられて多くのESの研究者が着目しているMulti-parent型の組換え操作[45][47]を第6章で扱う．

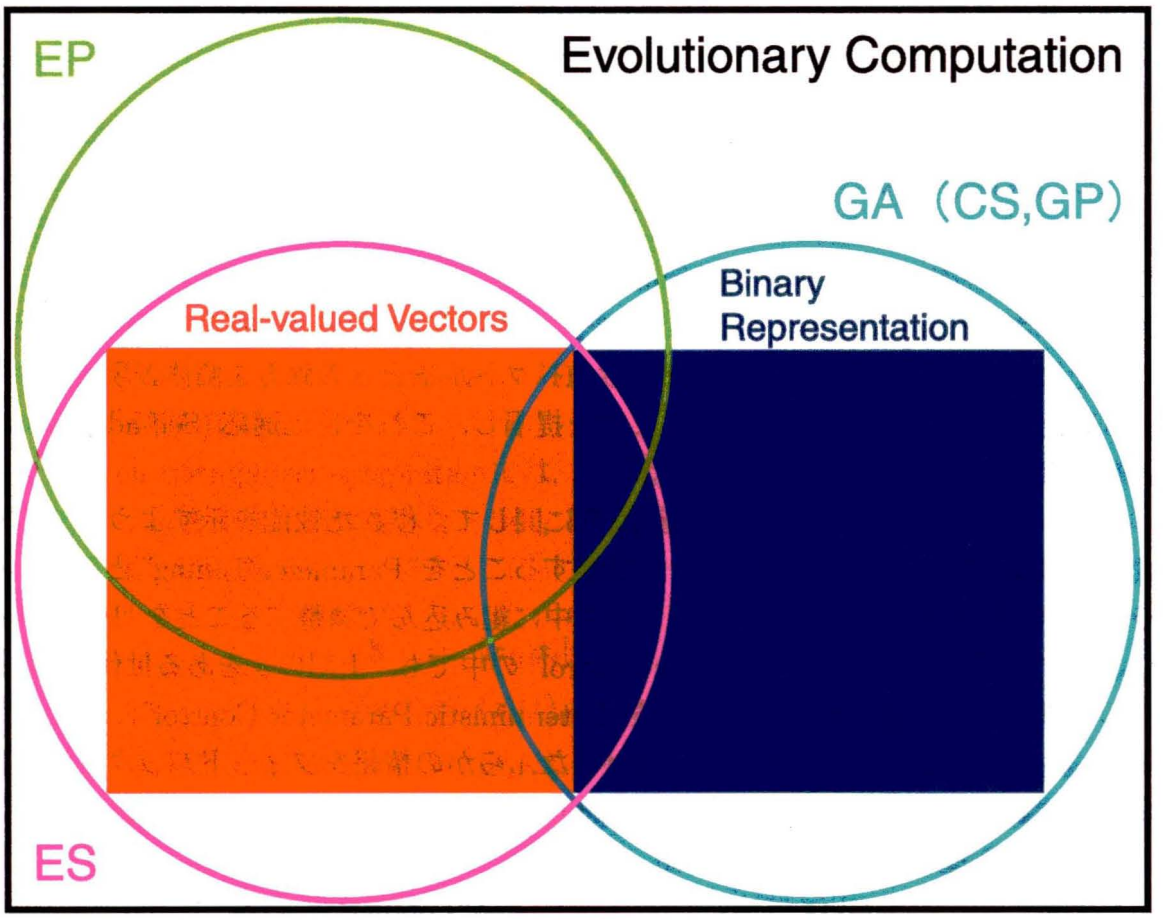
### 2.3.2 パラメータ設定と自己適応

標準的なGAはバイナリ表現を用いて，一点交叉と一点突然変異によって遺伝的操作を行い，ルーレット選択を行う．そのため，集団の大きさ，交叉率，突然変異率等のパラメータ設定が必要となる．パラメータ設定の際には多くの研究者が，数多くの計算機実験によって優れた性能を示すようパラメータ調整を行っている．過去の代表的なGA研究者の1人であるDe Jongの博士論文では，多くのテスト問題を解くことによって，集団サイズ50，交叉率0.6，突然変異率0.001等の推奨値を示している．また同様に，Grefenstette等，多くの研究者が，幅広い問題を扱い，一般的な問題に対する最適なパラメータ設定を発見しようとした．このことはGAの研究者のみならず，他の進化型計算でも同様に，一般的な問題に対する最適なパラメータ設定の発見が重要と考えられている．そのため，例えばESの研究者等は問題やアルゴリズム自身を単純化することによって理論的な解析を行っている．しかしながら，“任意の最適化問題に対する最適化能力の平均期待値は，どんな最適化アルゴリズムもランダム探索のそれと同じかそれよりも下回る”とするNFL定理(No-Free-Lunch Theorems) [40][111][112]が発表されると，一般の問題に対する最適なパラメータ設定の重要性が薄れ，個々の問題に適切なパラメータ設定を発見することや，その適切なパラメータ設定の調整法を発見することが重要と考えられるようになった．

一方，最適なパラメータを研究者が計算機実験を行って発見するのではなく，パラメータの調整機構をアルゴリズム自身の中に組み込むことが考えられてきた．例えば，$N(0,\sigma)$を平均0，標準偏差$\sigma$の標準正規分布から得られる乱数値とし，実数値ベクトル$\boldsymbol{x}$の突然変異を以下のように定義する．

$$\tilde{x}_i \;=\; x_i + N(0,\sigma) \tag{2.5}$$

探索の初期の段階では$\boldsymbol{x}$の変化量は比較的大きな変異が望ましく，探索の終盤では小



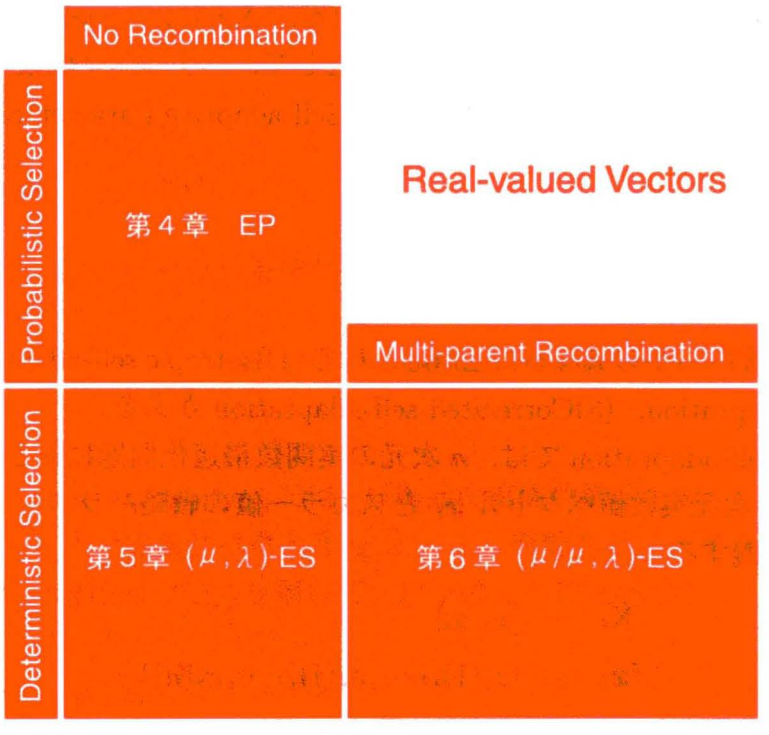




Fig. 2.1: The research map.

さな変異が望ましいため，$\sigma$ をある世代 $t$ の関数として

$$\sigma(t) = 1 - \alpha t \tag{2.6}$$

と定義することによって，世代に依存して動的に変化量を制御することが考えられた．ただし，$\alpha$ は定数とする．この方法の延長線上において，ES では，1973 年に Rechenberg によって"$\frac{1}{5}$ Success Rule" を用いた $\sigma$ の制御が提案された．これは，$\sigma$ をある世代 $t$ の関数とするのではなく，進化の過程において突然変異の成功確率が $\frac{1}{5}$ より大きくなると $\sigma$ を小さくし，$\frac{1}{5}$ より小さくなると $\sigma$ を大きくするというルールに基づくものであった．その後，1981 年には Schwefel が実数値ベクトル $\boldsymbol{x}$ と $\sigma$ の両方を個体表現に用いて $\sigma$ も進化させる手法として拡張することを提唱し，これを自己適応 (Self-adaptation) と呼んだ．

Eiben et al. [48][65] はパラメータ設定に関して，優れた性能を示すよう多くの事前の計算機実験によってパラメータを調整することを"Parameter Tuning" と呼び，パラメータの調整機構をアルゴリズム自身の中に組み込んで調整することを "Parameter Control" と呼んだ．この "Parameter Control" の中でも，上記の $\sigma$ をある世代 $t$ の関数として動的に変化量を制御する手法を"Deterministic Parameter Control"，また，"$\frac{1}{5}$ success rule" のように，進化過程におけるなんらかの情報をフィードバックさせて変化量を制御することを"Adaptive Parameter Control"，そして，$\sigma$ も個体表現に用いて変化量をも進化させることを"Self-adaptive Parameter Control" として分類している．また，Angeline [5] は "Parameter Control" を Population-level，Individual-level，Component-level の 3 つのレベルに分類している．さらに，Angeline の 3 つのレベルに，Hinterding et al. [65] は，4 つめのレベルとして，ペナルティ関数の重み等が属する Environment-level を提案している．なお，Self-adaptive Parameter Control は 4 つのレベルのうち Component-level に当てはまる．

### 2.3.3 自己適応の関連研究

ES には，代表的な 3 つの異なる自己適応として (1)Isotropic self-adaptation，(2)Non-isotropic self-adaptation，(3)Correlated self-adaptation がある．

(1)Isotropic self-adaptation では，$n$ 次元の実関数最適化問題に対し，個体 $\mathbf{X}_i (i = 1, 2, \cdots, \mu)$ を $n$ 次元実数値ベクトル $\boldsymbol{x}_i$ とスカラー値の戦略パラメータ $\sigma_i$ を用いて，以下のように定義する：

$$\mathbf{X}_i = [\boldsymbol{x}_i, \sigma_i] \tag{2.7}$$

$$^t\boldsymbol{x}_i = (x_i(1), \cdots, x_i(j), \cdots, x_i(n)) \quad (x_i(j) \in R, \sigma_i \in R^+) \tag{2.8}$$

ここで $j = 1, 2, \cdots, n$ であり，$x_i(j)$ はベクトル $\boldsymbol{x}_i$ の第 $j$ 成分である．Isotropic self-adaptation の突然変異では，$\sigma_i$ を用いて，$\boldsymbol{x}_i$ の変化量を規定している．これは，以下

のように定義されている.

$$\sigma_i^{(t+1)} = \sigma_i^{(t)} exp(\tau_0 N(0,1)) \tag{2.9}$$

$$x_i(j)^{(t+1)} = x_i(j)^{(t)} + \sigma_i^{(t+1)} N_i(0,1) \tag{2.10}$$

ただし，$\tau_0$ は定数であり，$N_i(0,1)$ は $i$ 成分毎に平均0，標準偏差 $\sigma$ の標準正規分布から得られる乱数値とする．この Isotropic self-adaptation に関しては，Beyer [24] によって (1,λ)-ES の Learning Parameter における Schwefel の $\tau$-scaling が証明されている．さらに，EP の Gaussian Perturbation が ES の $\sigma$-SA をテイラー展開したものと一致することを指摘している.

(2)Non-isotropic self-adaptation では，$n$ 次元の実関数最適化問題に対し，個体 $\mathbf{X}_i(i = 1,2,\cdots,\mu)$ を $n$ 次元実数値ベクトル $\boldsymbol{x}_i$ と $n$ 次元戦略パラメータベクトル $\boldsymbol{\sigma}_i$ を用いて，以下のように定義する：

$$\mathbf{X}_i = [\boldsymbol{x}_i, \boldsymbol{\sigma}_i] \tag{2.11}$$

$${}^t\boldsymbol{x}_i = (x_i(1),\cdots,x_i(j),\cdots,x_i(n)) \tag{2.12}$$

$${}^t\boldsymbol{\sigma}_i = (\sigma_i(1),\cdots,\sigma_i(j),\cdots,\sigma_i(n)) \tag{2.13}$$

$$(x_i(j) \in R, \sigma_i(j) \in R^+)$$

ここで $j = 1,2,\cdots,n$ であり，$x_i(j)$，$\sigma_i(j)$ はそれぞれベクトル $\boldsymbol{x}_i$，$\boldsymbol{\sigma}_i$ の第 $j$ 成分である．Non-isotropic self-adaptation の突然変異では $\boldsymbol{x}_i$ の $j$ 成分毎に，$\boldsymbol{\sigma}_i(j)$ が用いられ，以下のように定義されている.

$$\sigma_i(j)^{(t+1)} = \sigma_i(j)^{(t)} exp(\tau' N_i(0,1) + \tau N_{ij}(0,1)) \tag{2.14}$$

$$x_i(j)^{(t+1)} = x_i(j)^{(t)} + \sigma_i(j)^{(t+1)} N_{ij}(0,1) \tag{2.15}$$

ただし，定数 $\tau$ および $\tau'$ は，標準的な値として以下のように設定される [10]：

$$\tau = (\sqrt{2\sqrt{n}})^{-1} \tag{2.16}$$

$$\tau' = (\sqrt{2n})^{-1} \tag{2.17}$$

(3)Correlated self-adaptation では，$n$ 次元の実関数最適化問題に対し，個体 $\mathbf{X}_i(i = 1,2,\cdots,\mu)$ を $n$ 次元実数値ベクトル $\boldsymbol{x}_i$ と $n$ 次元戦略パラメータベクトル $\boldsymbol{\sigma}_i$ と回転角の $n \times n$ の共分散行列 $C(\boldsymbol{\alpha}_i)$ を用いて，以下のように定義する：

$$\mathbf{X}_i = [\boldsymbol{x}_i, \boldsymbol{\sigma}_i, C(\boldsymbol{\alpha}_i)] \tag{2.18}$$

$${}^tC(\boldsymbol{\alpha}_i) = (\boldsymbol{\alpha}_{i1},\cdots,\boldsymbol{\alpha}_{ij},\cdots,\boldsymbol{\alpha}_{in}) \tag{2.19}$$

$$(\boldsymbol{x}_i \in R, \boldsymbol{\sigma}_i \in R^+, 0 \leq \boldsymbol{\alpha}_{ij} \leq 180)$$

ここで$j=1,2,\cdots,n$であり，ベクトル$\boldsymbol{\alpha}_{ij}$は$n\times n$の共分散行列$C(\boldsymbol{\alpha}_i)$の第$j$行である．Correlated self-adaptationの突然変異では，以下のようにベクトル計算される：

$$\boldsymbol{\sigma}_i^{(t+1)} = \boldsymbol{\sigma}_i^{(t)} exp(\tau' N(0,1) + \tau N_i(0,1)) \tag{2.20}$$

$$\boldsymbol{\alpha}^{(t+1)} = \boldsymbol{\alpha}^{(t)} + \beta N_i(0,1) \tag{2.21}$$

$$\boldsymbol{x}_i^{(t+1)} = \boldsymbol{x}_i^{(t)} + \vec{N}(\vec{0}, C(\vec{\sigma}^{(t+1)}, \vec{\alpha}^{(t+1)})) \tag{2.22}$$

ここで，$\vec{N}(\vec{0}, C(\vec{\sigma}^{(t+1)}, \vec{\alpha}^{(t+1)}))$は平均ベクトルが0，共分散行列$C$の相関突然変異ベクトルを表し，$\beta$は定数である．Correlated self-adaptationに関して，Hansen [59]は共分散行列を取り扱うため計算時間が増え，しかも性能は初期値に大きく依存するために注意して初期値を選定しなければならないという欠点を述べている．

EPでは，自己適応の方式として，SaravananとFogel [101]が，従来の標準正規分布を用いたGaussian Perturbationより指数関数を導入したLog-normal Perturbationが有効であることを示し，GehlhaarとFogel [55]は，再生に関して突然変異させた戦略パラメータを用いるSigma-fast方式が有効であることを示している．これより現在では，Log-normal Self-adaptationを用いたSigma-fast方式が標準的手法となっている．これは，ESで用いられているNon-isotropic self-adaptationと同じである．また，実関数最適化問題だけでなく，組合せ最適化問題のTraveling Salesman Problemsにも適用され，例えば，ChellapillaとFogel [35]はInversionにおける個体長を自己適応させている．

GAでは，SchafferとMorishima [103]が交叉位置を，Bäck [9]が突然変異率を，Spears [108]が交叉率を，SmithとFogarty [107]は突然変異率と交叉率の両方を自己適応させている．また，Hinterding [63][64]はバイナリ表現にGauss型突然変異を加え，この時の変化量を自己適応させている．DebとBeyer [41][42]が，実数値GAに彼らが提案するSimulated Binary Crossover (SBX)を用いて，Self-adaptationと同様の特性を実現している．そして，適切なパラメータ設定を用いれば，このSBXはBlend crossover (BLX-$\alpha$)とFuzzy Recombination (FR)と集団の分散が同じになることを示している [26][28][30]．また，GPでは，Fogel et al. [54]が有限状態機械の5つの突然変異率を．Angeline [7]が交叉位置と交叉位置の数を自己適応させている．

本論文は，自己適応に関して，実関数最適化問題において優れた性能を示し，現在の標準的手法であるESのNon-isotropic self-adaptation (EPにおけるLog-normal Self-adaptationを用いたSigma-fast方式)に着目する．

### 2.3.4 自己適応の問題点

突然変異のステップサイズを制御する戦略パラメータは，その適応する性質によって適切に制御されると考えられている．しかし，問題が高次元になったり，適応度景観が複雑になるなどして最適化が困難になってくると，集団が大域最適解を発見する

以前に戦略パラメータが非常に小さく ($\sim 0$) なる傾向が現れ，その結果，突然変異のステップサイズも極めて小さくなり，実質的に探索点が動けない状態に陥り集団が収束する，いわゆる早期収束が起こりやすくなる．

一般には，これを回避するために，戦略パラメータの下限 $\epsilon$ を設定して小さくなり過ぎないようにするが，本論文の計算機実験でも示すように，探索性能は $\epsilon$ の値に大きく依存するとともに，適切な $\epsilon$ は問題の特徴によって異なるという結果が得られている．(なお，これと同様な現象は EP でも観測されている [73][74][75]．) そのため，戦略パラメータに個々の問題特性を考慮した上で下限を設定する必要がある．しかし，下限設定は (1) それ以上ステップサイズが小さくならず局所探索の能力を衰えさせてしまうことや，(2) 問題の特性に合わせた最適な下限を調節することは本来進化型計算が目指している頑健性の視点からは好ましくないことから，望ましい解決法とは言い難い．そこで，下限設定を行わずに戦略パラメータの適応能力を向上させる手法が望まれる．

## 2.4 結言

本章では，本論文において対象とする進化型計算手法のうち 1960 年代に始まった進化戦略，進化的プログラミング，遺伝的アルゴリズムの 3 つの手法の概略を紹介した．次に，本論文が取り扱う実関数最適化問題を定義し，この問題領域における自己適応を説明した．特に，各手法に関する自己適応の関連研究について述べ，研究領域を明らかにした．そして，探索空間の次元が高くなるほど，また，適応度景観が複雑になるほど，集団が大域最適解を発見する以前に戦略パラメータが非常に小さくなる傾向が現れ，その結果，突然変異のステップサイズも極めて小さくなり，実質的に探索点が動かない状態に陥って集団が収束することが自己適応の問題点であると指摘した．

# 第3章　自己適応の拡張と Multi-parent Recombination

## 3.1　緒言

本章では，本論文が動機付けられた生物進化メタファについて述べ，戦略パラメータの適応能力を向上させる拡張手法の基本的アイデアを説明する．そのアイデアは，分子進化の中立説に動機付けられて，戦略パラメータの進化に選択圧によらない確率的変化要因を加えることができるようにして，特定の値に固着しにくくしようとするところにある．そのために，各有効な戦略パラメータの他に冗長な戦略パラメータを持つ個体表現法を示し，冗長なものと有効なものを確率的に入れ替える新しい突然変異手法を定式化する．さらに，自己適応の拡張を補完する Multi-parent Recombination(MPR) について関連研究を述べ，本論文において参照する 3 つの MPR について説明する．

## 3.2　自己適応の拡張

### 3.2.1　ネオ・ダーウィニズム

Darwin の進化論で鍵となる要素は，偶然による遺伝的変異と適者生存の自然淘汰である．Darwin は遺伝的変異がどのようにして生じるかについては知らなかったが，いったん生じた遺伝的変異に，いかにして自然淘汰が働くかを明瞭にとらえていた．

そして，近年の「ネオ・ダーウィニズム」（neo - Darwinism）という言葉は，19 世紀末以来，ダーウィニズムの様々な種類の変形を表す言葉として使われてきた．現在，一般的には，Fisher，Wright，Haldane によって定式化された進化論を意味する．この三人は，1920 年代と 1930 年代に，突然変異，自然淘汰，および遺伝的浮動による遺伝子頻度の変化に関する数学的研究を精力的に行ない，遺伝子頻度を変化させるには，自然淘汰のほうが突然変異よりもはるかに効果的であるという結論に達した．そして，Dobzhansky の『遺伝学と種の起原』という著書を通して，ネオ・ダーウィニズムは生物学者の間に次第に受け入れられ，何人かの研究者 (Dobzhansky, Muller, Simpson, Huxley, Mayr, Stebbins, Ford 等) が，この理論を精巧なものとした．King に従って，ネオ・ダーウィニズムの特徴を述べると次のようになる．

1. 突然変異は遺伝子の働きとは関係無しに生じ，十分に高い頻度で再発する．

2. 突然変異は遺伝的変異の主要な供給源ではあるが，遺伝子頻度の変化に対する効果は小さいので，進化の上では脇役である．

3. 過去に生じた突然変異のため，自然集団にはほとんどいかなる種類の自然淘汰にも対応できるだけの十分な量の遺伝的変異が存在する．

4. 進化は主として環境の変化と自然淘汰によって決定される．すでに十分な量の遺伝的変異が存在するので，環境の変化に反応して集団が進化するためには，新しい突然変異は必要とされない．突然変異率と進化速度には関係が無い．

5. 突然変異は十分に高い頻度で再発する傾向があるので，明らかに生存に有利な突然変異のほとんどは，すでに集団内で固定しているか，あるいは最適頻度に達しているはずである．従って，集団の遺伝的構成は，与えられた環境の中では，ほとんどいつも最適状態か，あるいはそれに近い状態となっている．

6. 種の進化的変化は自然淘汰によって徐々に起こる．従って，大進化は小進化の効果が蓄積したものに過ぎない．

Ford は生物のほとんどの進化的変化が強い自然淘汰によって生じるという見解をとり，一方，Simpson は新しい適応形質の創造における突然変異の役割を重視した．Wright は 1931 年と 1932 年に発表した論文の中で，平衡転移による進化理論を提唱した．これは，分割された小集団内で生じる自然淘汰と遺伝的浮動を複合させた効果によって，集団全体の平均適応度が増加するというものである．Dobzhansky と Simpson はこの理論を受け入れたのみならず，種より高いレベルでの進化を説明することにも用いた．

しかし，ネオ・ダーウィニズム論者の間には，遺伝的多型の保持機構に関して重要な意見の不一致が存在した．Dobzhansky，Ford，Wallace によって代表される学派は，生存力と繁殖力に影響する遺伝的多型の大部分が，ある種の平衡淘汰によって維持されていると考えた．一方，Muller と Crow に代表される学派は，突然変異と自然淘汰との平衡によって遺伝的多型が維持されていると考えた．Dobzhansky は，前者を平衡説，後者を古典説と呼んだ．そして，古典説は，Morgan の突然変異 - 自然淘汰説と本質的に同一である．1950 年代から 1960 年代にかけて，これら二学派の間で論争がなされた．特に，ホモ接合体において有害な放射線で誘発された突然変異が，ヘテロ接合体でどういう効果を生じるかが，大きな問題点のひとつとなった．これは，のちにタンパク質多型の維持機構をめぐる論争につながる．

観察事実として，以下が知られている．(1) 各々のタンパク質ないし遺伝子において，アミノ酸や塩基の置換数を用いて測られた進化速度は，その遺伝子の機能が変わらない限り，様々な生物の系統において，年あたりほぼ一定である．(2) 機能的にそれ

ほど重要でない遺伝子 (あるいはその一部) は，より重要な遺伝子よりも速く進化する．しかし，これらの観察事実はネオ・ダーウィニズムの原理の大部分と矛盾することが分かる．ネオ・ダーウィニズムにおいては，進化速度は環境がいかに頻繁に，いかに速く変化するかに依存する．同様に，機能的にあまり重要ではない遺伝子あるいは遺伝子の一部分が，機能的に重要なものより速く進化するという観察事実も，ネオ・ダーウィニズムとは折り合わない．ネオ・ダーウィニズムにおいては，ほとんどの遺伝子置換は正のダーウィン流淘汰によって生じるものとされ，従って機能的に重要な遺伝子あるいはその一部は，機能上さほど重要でないものよりも，進化速度が大きいと期待される．ネオ・ダーウィニズムに基づいた数学モデルによって，タンパク質とＤＮＡの遺伝的多型の程度と，塩基の置換速度の両方を満足に説明することもまた困難である．さらに，高等生物のゲノムは柔軟性に富み，表現型にあまり影響を与えることなく，かなり急速な構造変化を起こすことができる．

### 3.2.2 分子進化の中立説

Kimuraは分子レベルでの進化的変化，すなわち遺伝物質それ自身の変化を引き起こす主な要因として正のダーウィン淘汰では分子進化をうまく説明することができないと気付き，分子進化の中立説 [70] を提唱した．中立説とは，分子レベルにおける進化的変化と多型は主に自然淘汰に関してほとんど中立で，その行動と運命が主として突然変異と遺伝的浮動によって決定されるような突然変異遺伝子によるものであるとする説である．“中立説” とは中立性それ自体を強調しているのではなく，主な説明要因としての突然変異と遺伝的浮動を強調するものである．そのため，淘汰が何の役割も果たしていないと考えられているわけではない．しかし，分子変化の相当の部分がネオ・ダーウィニズムに基づく正の淘汰によることや，分子多型が平衡淘汰によって決定されているということは否定されている．この理論は，その後多くの研究者によって洗練されてきた．現在の中立説は，以下のように特徴付けられている [87].

1. アミノ酸と塩基の置換は，そのほとんどが中立もしくは中立に近い突然変異の偶然による固定によって生じる．

2. 集団内の遺伝的多型は，遺伝子置換の過程の一断面である．

3. 中立説は，進化の過程で集団中に取りこまれる大多数の突然変異の動態に関する理論であり，生存に有利な突然変異や超優性を示す突然変異が少数存在することを否定するものではない．

4. ほとんどの新生突然変異は有害であり，集団から急速に失われると仮定される．よって，それら有害な突然変異は，集団内の遺伝的変異や遺伝子の置換にはあまり寄与しない．

5. 中立な対立遺伝子は機能の無い遺伝子ではなく，一般には生物の生存にとって不可欠な重要性を持っている．2個の対立遺伝子が機能的に同等であり，生物の適応度の大小に影響を与えなければ，“中立である”(neutral) という．しかし集団遺伝学では，2個の対立遺伝子が中立であるかどうかは，集団におけるその動態がどの程度まで遺伝的浮動によるかによって定義される．従って，大集団では有利である突然変異遺伝子も，小集団では中立になりうる．

### 3.2.3 中立説に動機付けられた自己適応の拡張

2章で指摘したように，進化過程における戦略パラメータの適応能力が十分でないと考え，これを向上させることにより探索の頑健性を得ることを試みる．ECを最適化問題に適用する際は，集団の収束が早すぎれば大域最適解を発見できなくなってしまう一方，逆に，遅すぎれば非常に大きな計算コストがかかってしまうため，再生と選択のバランスをうまくとって多様性の制御に注意しなければならない．特に，ECの探索点集団は自然選択の効果で次第に収束していくが，各個体の探索範囲を決定する戦略パラメータが進化の過程で0に近づいてしまい，突然変異のステップサイズが小さくなりすぎることが，自己適応の問題点の本質であると考えられる．そこで，分子進化の中立説 [70](3.2.2の1) に動機付けられて，戦略パラメータの変異に遺伝的浮動の効果を導入して，適応能力の向上をはかる．

この基本的アイデアは，戦略パラメータが探索の比較的早い段階で0に近くなり，集団が実質的に動かなくなる早期収束の状態に陥るという観察事実に着目し，これを避けるため戦略パラメータの進化に選択圧によらない確率的変化要因を加えることができるようにして，特定の値に固着しにくくしようとするところにある．それゆえ，各有効な戦略パラメータの他に各個体の目的関数に対して中立な領域として，冗長な戦略パラメータを持つ個体表現法を用い，冗長なものと有効なものを確率的に入れ替える新しい突然変異手法を導入する．なお，個体変化に選択圧によらない変化，すなわち，遺伝的浮動を用いる基本的な着想は，冗長な遺伝子型とそれを有効活用する遺伝的操作を用いて，実効突然変異率を適応的に変化させることにより多様性を制御するGA (operon-GA [88]) から来ている．

### 3.2.4 個体表現の拡張

$n$次元の実関数最適化問題に対し，個体 $\mathbf{X}_i(i = 1, 2, \cdots, \mu)$ を$n$次元実数値ベクトル $\boldsymbol{x}_i$ と$n$次元戦略パラメータベクトル $\boldsymbol{\eta}_{i0}$ および，冗長な$m$個の$n$次元戦略パラメー

タベクトル $\boldsymbol{\eta}_{il}(l=1,2,\cdots,m)$ を用いて，以下のように定義する：

$$\begin{aligned}
\mathbf{X}_i &= [\boldsymbol{x}_i,(\boldsymbol{\eta}_{i0},\cdots,\boldsymbol{\eta}_{ik},\cdots,\boldsymbol{\eta}_{im})] && (3.1)\\
{}^t\boldsymbol{x}_i &= (x_i(1),\cdots,x_i(j),\cdots,x_i(n)) && (3.2)\\
{}^t\boldsymbol{\eta}_{ik} &= (\eta_{ik}(1),\cdots,\eta_{ik}(j),\cdots,\eta_{ik}(n)) && (3.3)\\
&\quad (x_i(j)\in R,\eta_{ik}(j)\in R^+)
\end{aligned}$$

ここで $j=1,2,\cdots,n$，$k=0,1,\cdots,m$ であり，$x_i(j)$，$\eta_{ik}(j)$ は，それぞれベクトル $\boldsymbol{x}_i$，$\boldsymbol{\eta}_{ik}$ の第 $j$ 成分である．個体 $\boldsymbol{X}_i$ は $n\times m$ 個の戦略パラメータを持つが，子孫を生成する際には一番目の要素 $\boldsymbol{\eta}_{i0}$ のみしか使わない．それゆえ，$\boldsymbol{\eta}_{i0}$ を活性戦略パラメータ，それ以外の冗長な $\boldsymbol{\eta}_{il}(l=1,\ldots,m)$ を不活性戦略パラメータと呼ぶことにする．

### 3.2.5 突然変異の拡張

まず，戦略パラメータベクトルの各要素に，以下に定める三つの操作 $(g_{dup}, g_{del}, g_{inv})$ を確率的に実施する：

$$\begin{aligned}
g_{dup}: \quad & \eta'_{i0}(j)=\eta_{i0}(j) && (3.4)\\
& \eta'_{il}(j)=\eta_{i(l-1)}(j)\\
& \tilde{\eta}_{ik}(j)=D(\eta'_{ik}(j))\\
g_{del}: \quad & \eta'_{i(l-1)}(j)=\eta_{il}(j) && (3.5)\\
& \eta'_{im}(j)=min(\eta_{max},\sum_{r=1}^{m}\eta_{ir}(j))\\
& \tilde{\eta}_{ik}(j)=D(\eta'_{ik}(j))\\
g_{inv}: \quad & \eta'_{i0}(j)=\eta_{il_0}(j) && (3.6)\\
& \eta'_{il_0}(j)=\eta_{i0}(j)\\
& \tilde{\eta}_{i0}(j)=D(\eta'_{i0}(j))\\
& \tilde{\eta}_{il_0}(j)=D(\eta'_{il_0}(j))
\end{aligned}$$

ここで，$l_0\in\{1,2,\cdots,m\}$ であり，戦略パラメータの上限 $\eta_{max}$ は定数であり，D は以下の式を表す：

$$D(\tilde{\eta}_{ik}(j)) = \eta'_{ik}(j)\times\exp(\tau' N(0,1)+\tau N_{ijk}(0,1))$$

ただし，$N_{ijk}(0,1)$ は $i,j,k$ ごとに得られるそれぞれ独立な標準正規分布に従う乱数値である．

また，これらの戦略パラメータに対する操作の意味を，以下のように解釈することができる．

- $g_{dup}$ ··· 現在の戦略 $\eta_i(j,1)$ を記憶する．そして，一番古く記憶されたと思われる戦略 $\eta_i(j,m)$ を捨てる．

- $g_{del}$ ··· 現在の戦略 $\eta_i(j,1)$ を捨て，ごく最近有効であった戦略 $\eta_i(j,2)$ を使う．そして，新しい戦略を記憶域に導入する．

- $g_{inv}$ ··· 現在の戦略 $\eta_i(j,1)$ を記憶し，記憶されている戦略 $\eta_i(j,p)$ を再利用する．

結果として，記憶域には以前有効であった戦略が記憶されるが，記憶されている世代数が短いほど再利用される率が高くなる．また，記憶されている世代数が長ければ長いほど，記憶当時の戦略を失っていく．

これらの機構により，各個体の戦略パラメータの大きさ分布が集団への淘汰圧に関係のない遺伝的浮動によって変化できるようになる．そのため，例えば，ある期間に活性戦略パラメータが小さくなり過ぎていても，遺伝的浮動によって高速に大きくなり，また逆に，ある期間に大きくなり過ぎていても，高速に小さくなることができる．なお，(3.5) 式の第 2 式では，不活性域の端に新しく挿入される値を規定している．なお，この値は基本的にどんな値でも設定可能ではあるが，他の不活性戦略パラメータとそれほどかけ離れた値にならないように設定する．

子孫の実数値ベクトルは，以下の式により親から再生する：

$$\tilde{x}_i(j) \quad = \quad x_i(j) + \tilde{\eta}_{i0}(j)\delta_{ij} \tag{3.7}$$

ここで，$\delta_{ij}$ は Cauchy 分布から得られる乱数値である．

$x_i(j)$ は $\eta_{i0}(j)$ のみで定まるため，$\eta_{il}(j), l = 1, \cdots, m$ における変化は目的関数に対して中立である．なお，各操作の適用確率 $P(g_*)$ をそれぞれ $P(g_{dup}) = 1.0$，$P(g_{del}) = 0.0$，$P(g_{inv}) = 0.0$ とすると，不活性戦略パラメータが活性化されることがなくなるため，ES の Non-isotropic self-adaptation (EP における Log-normal Self-adaptation を用いた Sigma-fast 方式) と全く等価になる．

## 3.3 Multi-parent Recombination

組換え操作の効果に関しては数理的な解析が困難なため，実験的に調べられている．このことは，Multi-parent Recombination(MPR) についても同様である．MPR に関する研究は，GA と ES の両分野で多数なされている．例えば GA では，Mühlenbein と Voigt [85] が Gene Pool Recombination を導入し，Discrete Binary Function に対する数理的な解析を報告している．また，Binary Functions [43] と NK-landscapes [44] を用いた計算機実験によって Scanning Crossovers と Diagonal Crossover に関して報告されている．

ES では，$\varrho$ 個 ($2 \leq \varrho \leq \mu$) の親個体から 1 個の子孫を生成する ($\mu/\rho$,$\lambda$)-ES が [105] において提案されている．そして，Beyer [23] は ($\mu/\rho$,$\lambda$)-ES に関する議論を整理し，

Sphere Function における $\rho = \mu$ の場合に，組換え操作のない場合と比べて収束率が $\mu$ 倍速いことを理論的に証明した．しかし，Beyer の理論的な考察では，戦略パラメータの組換え操作を考慮していない．また，Eiben と Bäck [46][15] は子孫を生成する時の親の数が $(\mu/\rho,\lambda)$-ES の性能に与える影響を実験的に示している．

$(\mu/\rho,\lambda)$-ES には，MPR [45][47] として Intermediate Recombination (IR) と Discrete Recombination (DR) があるため，実数値変数と戦略パラメータのどちらか，もしくは両方に対して用いることができる．Schwefel [104][8] 等，性能は問題に依存するため試行錯誤することを勧めながら，一般的には DR を実数値変数に，IR を戦略パラメータに用いることを推奨している．そのため，Eiben と Bäck [46][15]，Nissen と Propatch [86]，Gruenz と Beyer [57]，Beyer と Arnold [27][1]，Fogel と Beyer [52] 等，多くの ES の研究者は戦略パラメータには IR を用いる．しかしながら，Chang et al. [34] は $(\mu/\mu,\lambda)$-ES において戦略パラメータに IR を用いるのみならず，戦略パラメータに DR を用いる手法の有効性を計算機実験により明らかにした．この研究では，11 個の標準的なテスト関数によって，Classical-ES (CES) の性能比較を行っている．

そこで本論文では，Chang et al. [34] にならい，$(\mu/\mu,\lambda)$-ES で用いる MPR として，IR を実数値変数と戦略パラメータの両方に用いる Multi-parent Intermediary Recombination を採用する．さらに，DR を実数値変数と戦略パラメータの両方に用いる手法として，Multi-parent Discrete Recombination と Global Combined Discrete Recombination を採用する．ここで，DR を実数値変数と戦略パラメータに用いる場合，Multi-parent Discrete Recombination のように実数値変数と戦略パラメータを独立に取り扱うことができる．しかし，戦略パラメータによって実数値変数は規定されるため，戦略パラメータと実数値変数には相関があると考えられる．そこで，実数値変数と戦略パラメータを一組とみなして DR を用いる手法を Global Combined Discrete Recombination として取り扱う [34]．

### 3.3.1 Multi-parent Intermediary Recombination

Multi-parent Intermediary Recombination では，実数値変数と戦略パラメータの両方に IR を適用するため，子孫は親集団の中心から生成される．これは以下のように定式化される：

$$\tilde{\eta}_{ik}(j) = \frac{1}{\mu}\sum_{i=1}^{\mu}\eta_{ik}(j) \tag{3.8}$$

$$\tilde{x}_i(j) = \frac{1}{\mu}\sum_{i=1}^{\mu}x_i(j) \tag{3.9}$$

ここで，$i = 1, 2, \ldots, \mu$，$j = 1, 2, \ldots, n$，$k = 0, 1, \ldots, m$ である．なお，本論文ではこの操作を用いる ES を II-ES として参照する．

### 3.3.2 Multi-parent Discrete Recombination

Multi-parent Discrete Recombination では，子孫の実数値変数と戦略パラメータの第 $j$ 成分は集団内のあるランダムに選ばれた個体の第 $j$ 成分からそれぞれ生成される．これは以下のように定式化される：

$$\tilde{\eta}_{ik}(j) = \eta_{\chi_j k}(j) \tag{3.10}$$

$$\tilde{x}_i(j) = x_{\chi'_j}(j) \tag{3.11}$$

ここで，$\chi_j$ と $\chi'_j$ は，一様分布 $\{1 \ldots \mu\}$ から第 $j$ 成分ごとに実数値変数と戦略パラメータで独立に得られる整数乱数である．なお，本論文ではこの操作を用いる ES を DD-ES として参照する．

### 3.3.3 Global Combined Discrete Recombination

Global Combined Discrete Recombination では，子孫の実数値ベクトルと戦略パラメータを一組とみなし，それらの第 $j$ 成分は集団内のあるランダムに選ばれた個体の第 $j$ 成分から生成される．これは以下のように定式化される：

$$\tilde{\eta}_{ik}(j) = \eta_{\chi_j k}(j) \tag{3.12}$$

$$\tilde{x}_i(j) = x_{\chi_j}(j) \tag{3.13}$$

なお，本論文ではこの操作を用いる ES を D-ES として参照する．

## 3.4 結言

本章では，本論文において動機付けられている生物進化メタファについて述べ，戦略パラメータの適応能力を向上させる拡張手法の基本的アイデアを説明した．提案手法は，分子進化の中立説 [70] に動機付けられており，戦略パラメータの進化に選択圧によらない確率的変化要因を加えることで，特定の値に固着しにくくしようとするものであることを説明した．そして，各有効な戦略パラメータの他に冗長な戦略パラメータを持つ個体表現法を採用し，冗長なものと有効なものを確率的に入れ替える新しい突然変異手法を定式化した．さらに，自己適応の拡張を補完する Multi-parent Recombination(MPR) について関連研究を述べ，本論文において参照する 3 つの MPR について説明した．

# 第4章　EPにおける自己適応の拡張

## 4.1　緒言

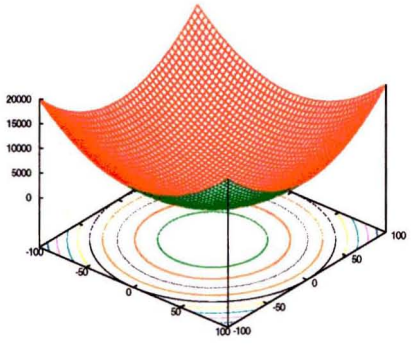
本章では，進化戦略の一種として考えられる進化的プログラミング (Evolutionary Programming：EP) に関する自己適応の拡張法について議論する．本章では，進化的プログラミングにおいて従来の Gauss 型突然変異を用いる手法を Classical-EP (CEP)，Cauchy 型突然変異を用いる手法を Fast-EP (FEP)，提案する新しい EP の拡張法を Robust-EP (REP) として参照する．本章の計算機実験では，実関数最適化問題において一般に用いられている標準テスト関数によって従来手法 (CEP，FEP) と提案手法 (REP) を比較検証し，自己適応の問題点に対する提案手法の有効性を明らかにし，進化ダイナミクスの解析により手法の特性を明らかにする．また，工学問題に応用する上で重要となるノイズに対する頑健性を検証した後に，実問題に応用しその有用性を検証する．特に，工学的実問題への応用として多指ハンドよる物体の把持問題を取り上げ，Liapunov の安定条件及び接触安定条件を満たす指先位置・指先力計画の最適化を行う．この問題は 27 の非線形制約条件付き最適化問題であり，超多峰性問題の一つと考えられる．特に，進化的プログラミングの特徴である最良個体が生存する点は，全ての拘束条件を一度充足すれば，少なくとも 1 個体が拘束条件を充足したまま生存することが約束されるため，この問題に有効であると考えられる．

Table. 4.1: Contents of Section 4.

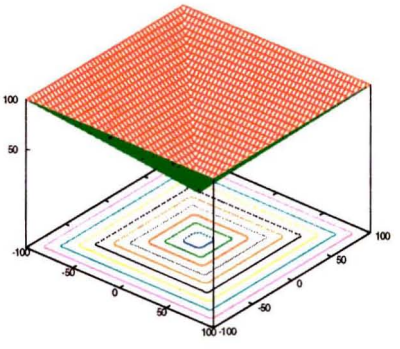
| 4章 | 4.2 | 4.3 | 4.4 |
|---|---|---|---|
| 問題 | 標準テスト関数 | ノイズを含むテスト関数 | 多指ハンドの把持計画 |
| 手法 | CEP,FEP,REP | CEP,FEP,REP | CEP,REP |

Table. 4.2: Eleven test functions.

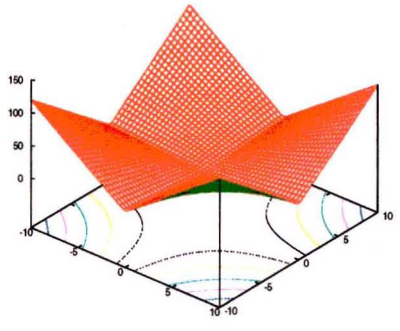
| Name | Expression($n = 30$) | Range |
|---|---|---|
| Hypersphere | $f_1(x) = \sum_{i=1}^{n} x_i^2$ | $-100 \leq x_i \leq 100$ |
| Schwefel's Problem 2.22 | $f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$ | $-10 \leq x_i \leq 10$ |
| Schwefel's Problem 1.2 | $f_3(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$ | $-100 \leq x_i \leq 100$ |
| Schwefel's Problem 2.21 | $f_4(x) = max\{\lvert x_i \rvert, 1 \leq x_i \leq n\}$ | $-100 \leq x_i \leq 100$ |
| Rosenbrock | $f_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$ | $-30 \leq x_i \leq 30$ |
| Step Function | $f_6(x) = \sum_{i=1}^{n} (\lfloor x_i + 0.5 \rfloor)^2$ | $-100 \leq x_i \leq 100$ |
| Rastrigin | $f_7(x) = \sum_{i=1}^{n} \{x_i^2 - 10\cos(2\pi x_i) + 10\}$ | $-5.12 \leq x_i \leq 5.12$ |
| Ackley | $f_8(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos 2\pi x_i\right) + 20 + e$ | $-32 \leq x_i \leq 32$ |
| Griewank | $f_9(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1$ | $-600 \leq x_i \leq 600$ |
| Penalized Function P8 | $f_{10}(x) = \frac{\pi}{n}\{10sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2 \times [1 + 10sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{n-1} u(x_i, 10, 100, 4)$ <br> $where, \quad y_i = 1 + \frac{1}{4}(x_i + 1)$ <br> $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \leq x_i \leq a \\ k(-x_i - a)^m & x_i < -a \end{cases}$ | $-50 < x_i < 50$ |
| Penalised Function P16 | $f_{11}(x) = 0.1\{sin^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i - 1)^2 \times [1 + sin^2(3\pi x_{i+1})] + (x_n - 1)^2[1 + sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$ | $-50 < x_i < 50$ |



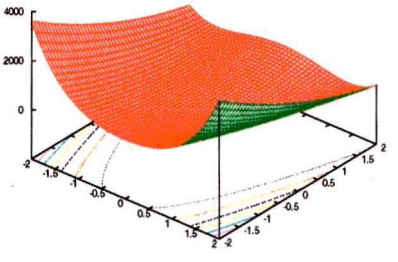

(a) $f_1$.



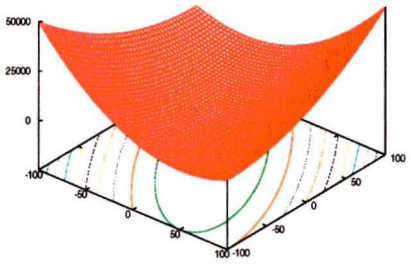

(d) $f_4$.



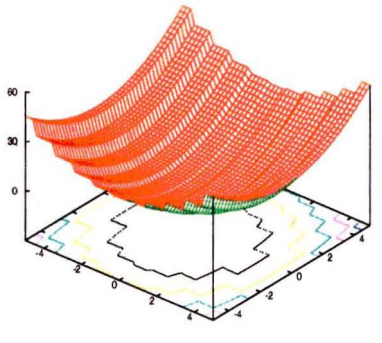

(b) $f_2$.




(e) $f_5$.

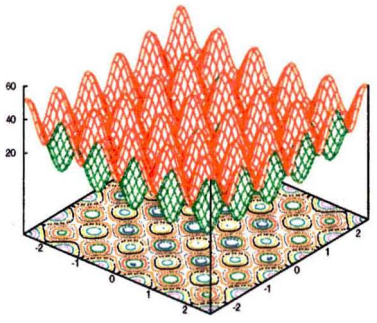



(c) $f_3$.

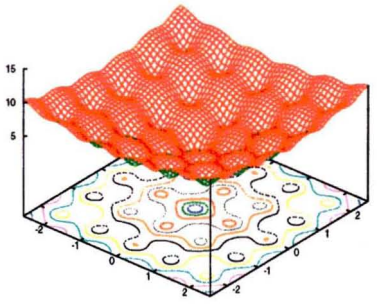



(f) $f_6$.

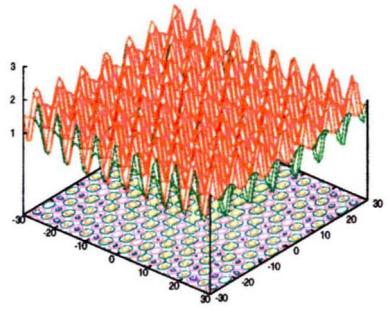
Fig. 4.1: The 2-dimensional landscapes on unimodal functions.



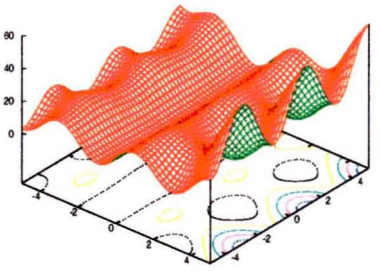
(a) $f_7$.



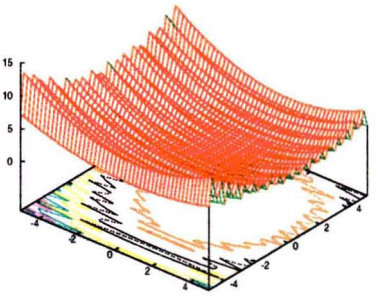
(b) $f_8$.



(c) $f_9$.



(d) $f_{10}$.



(e) $f_{11}$.

Fig. 4.2: The 2-dimensional landscapes on multi-modal functions.

## 4.2 標準テスト関数を用いた計算機実験

### 4.2.1 テスト関数と諸設定

REP の有効性と進化過程の特徴を明らかにするために，従来手法として CEP と FEP を取り上げ，Table.4.2 に示す 11 のテスト関数を用い計算機実験を行った．これらは，それぞれ Hypersphere 関数 ($f_1$)，Schwefel の問題 2.22($f_2$)，Schwefel の問題 1.2($f_3$)，Schwefel の問題 2.21($f_4$)，Rosenbrock 関数 ($f_5$)，Step 関数 ($f_6$)，Rastrigin 関数 ($f_7$)，Ackley 関数 ($f_8$)，Griewank 関数 ($f_9$) である．また，$f_1$ から $f_6$ までは単峰性，$f_7$ から $f_{11}$ は多峰性関数である (Fig. 4.1, Fig. 4.2)．すべてのテスト関数は 30 次元の探索空間を持ち，大域最小値は 0 である．

CEP と FEP および REP は，それぞれ [116][118][120][121] に従って，$\mu = 100$，トーナメントサイズ $q = 10$ として計算する．戦略パラメータの上限 $\eta_{max}$ は探索範囲が比較的狭い $f_7$ のみ 1.0，その他では 3.0 とし，下限 $\epsilon$ を $10^{-2}$，$10^{-4}$，$10^{-6}$，$10^{-8}$，$10^{-10}$ と 5 つの値に変えて，それぞれ 50 回試行を繰り返した．さらに，REP では，$m = 5$ として，$g_{dup}$，$g_{del}$，$g_{inv}$ の適用確率 (P) を，それぞれ $P(g_{dup}) = 0.6$，$P(g_{del}) = 0.3$，$P(g_{inv}) = 0.1$ とした [76]．なお，CEP と FEP の実験は，REP で CEP と FEP に等価になるパラメータ値，すなわち，$P(g_{dup}) = 1$，$P(g_{del}) = 0$，$P(g_{inv}) = 0$ として行った．

REP におけるこれらの推奨パラメータ値は，予備的な実験結果から決定した．また，同実験からパラメータ値の変化に関して REP は非常に頑健であることが分かったため，パラメータ値チューニングにはそれほど注意を払っていない．それゆえ，詳細な実験を通してこれらを決定すれば，パフォーマンスは更に向上すると思われる．

### 4.2.2 基本探索性能

#### • 計算機実験結果 (単峰性関数)

単峰性関数 $f_1$ から $f_6$ の計算結果を Fig. 4.3 から Fig. 4.8 に示す．それぞれ，縦軸に関数値，横軸に世代をとった．

CEP と FEP による $f_1$ の 50 回の平均結果を示した Fig. 4.3 より，CEP，FEP ともに下限が $10^{-4}$ より小さい場合，200 世代付近で早期収束の傾向が顕著に見られ収束速度が落ちている．一方，REP においては，下限によってそれほど収束速度の差異はなく，小さい下限を用いるほど精度良く大域最適解に近づくことができている．

そして，これらの傾向は $f_2$ から $f_4$ でも見られる．$f_2$ では下限が $10^{-4}$ より小さい場合に 200 世代付近で，$f_3$ では下限が $10^{-4}$ より小さい場合に 500 世代付近で，$f_4$ では下限が $10^{-4}$ より小さい場合に 1000 世代付近で，$f_5$ では下限が $10^{-4}$ より小さい場合に 500 世代付近で，早期収束の傾向が顕著に見られ収束速度が落ちている．そのため，下限が小さくなるにしたがって REP の優位性が現れている．

とくに，$f_6$ の結果を示した Fig. 4.8 から $f_1$ から $f_5$ で見られる CEP と FEP と REP

Table. 4.3: Summary of experiments for standard test functions.

| | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ | $f_8$ | $f_9$ | $f_{10}$ | $f_{11}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| CEP | × | × | × | × | × | × | × | × | × | × | × |
| FEP | × | × | × | × | × | × | × | × | × | × | × |
| REP | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |

の探索過程の違いはいっそう明確に現れる．CEPは全く最適解0を発見できず，FEPは$\epsilon = 10^{-2}$の場合に大域最適解に到達しているものの，$10^{-4}$より小さくなってしまうと最適解を発見できていないことが分かる．一方，REPでは，すべての下限で最適解に到達していることが分かる．

### • 計算機実験結果 (多峰性関数)

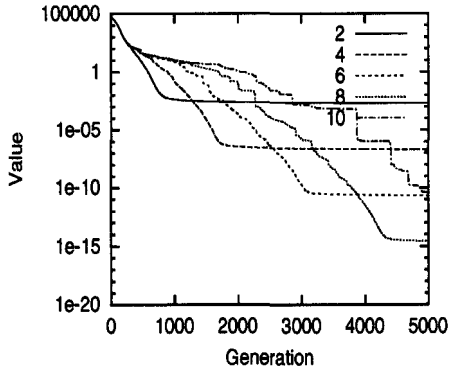
次に，多峰性関数$f_7$から$f_{11}$の計算結果をFig. 4.9からFig. 4.11に示す．それぞれ縦軸に関数値，横軸に世代をとった．

CEPとFEPによる$f_7$の結果を示したFig. 4.9より，CEP，FEPともに下限が$10^{-4}$より小さい場合，100世代付近で早期収束の傾向が顕著に見られ収束速度が落ちている．また，FEPにおいて下限が$10^{-4}$の場合にやや改善が見られるもののその他ではFEP，CEPとも明らかに集団が局所解から抜け出せない傾向も示している．一方，REPにおいては，単峰性関数の傾向と同様，下限の違いによる収束速度の差異もほとんどなく，小さい下限を用いるほど精度良く大域最適解に近づくことができている．

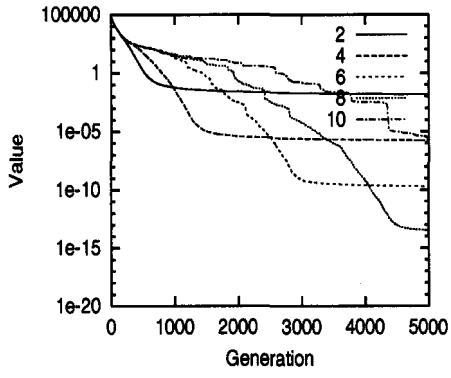
そして，これらの傾向は，$f_8$から$f_{11}$でも見られる．$f_8$ではCEPとFEPの下限が$10^{-4}$より小さい場合に300世代付近で，$f_9$ではCEPの下限が$10^{-4}$より小さい場合に1100世代付近で，FEPの下限が$10^{-4}$より小さい場合に300世代付近で，$f_{10}$ではCEP，FEPの下限が$10^{-4}$より小さい場合に300世代付近で，$f_{11}$ではCEP，FEPの下限が$10^{-4}$より小さい場合に100世代付近で，早期収束の傾向が顕著に見られ収束速度が落ち，局所解から抜け出せていない．一方，REPにおいては，下限の違いによる収束速度の差異もほとんどなく，小さい下限を用いるほど精度良く大域最適解に近づくことができている．

### • 計算機実験結果 (まとめ)

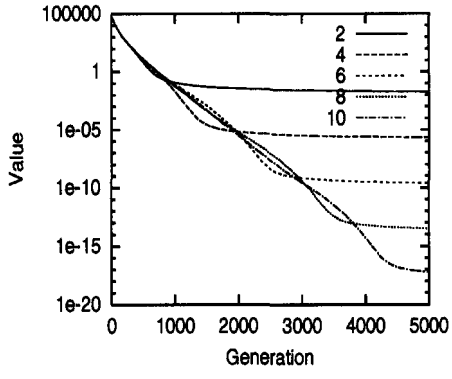
計算機実験結果によって得られた下限設定に対する頑健性をTable.4.3にまとめる．CEPとFEPは，全てのテスト関数において下限$\epsilon$によって異なる収束速度を示す．一方，REPは$\epsilon$の設定に依存せず，頑健であった．




(a) CEP.

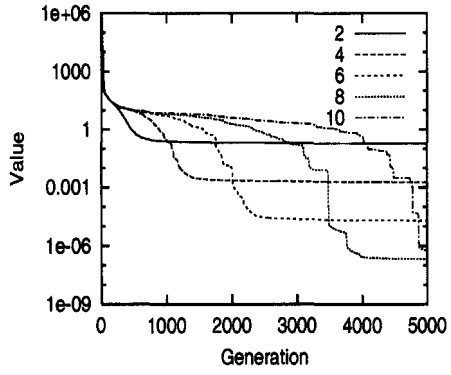



(b) FEP.

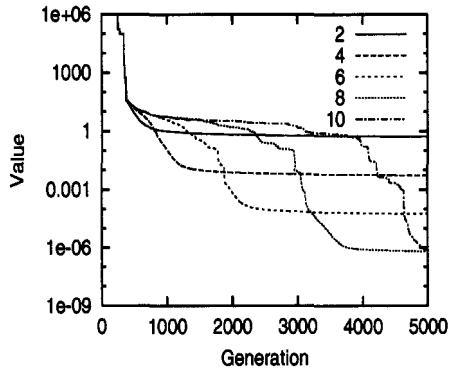



(c) REP.

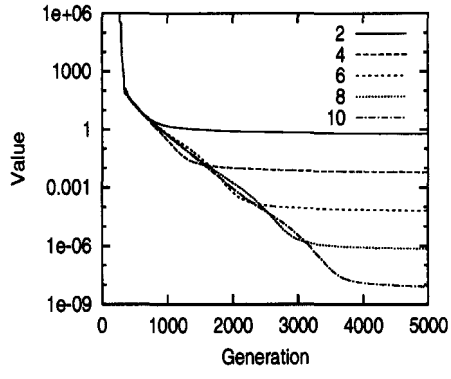
Fig. 4.3: Averaged best results on $f_1$ when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.




(a) CEP.

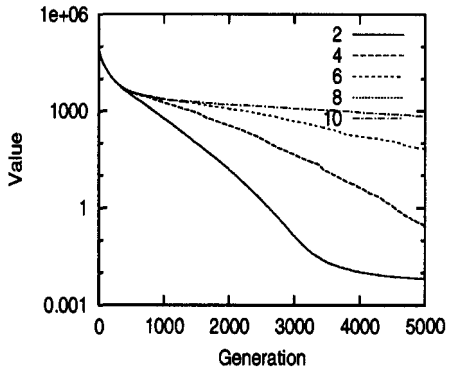



(b) FEP.

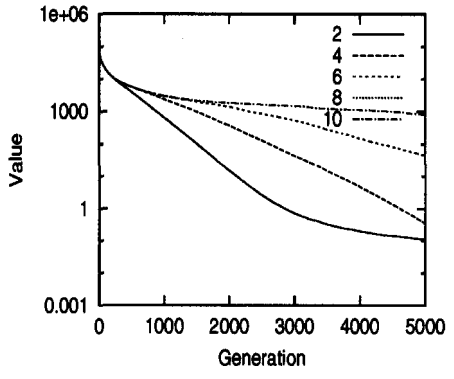



(c) REP.

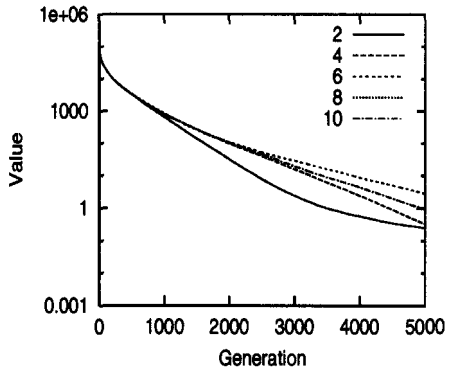
Fig. 4.4: Averaged best results on $f_2$ when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.




(a) CEP.

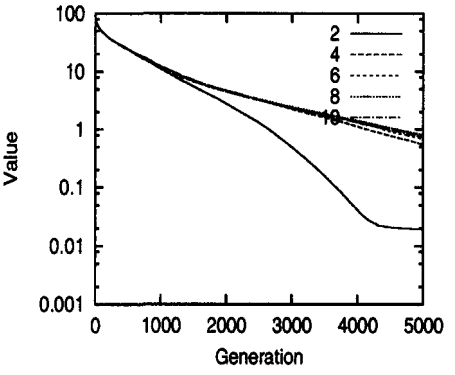



(b) FEP.

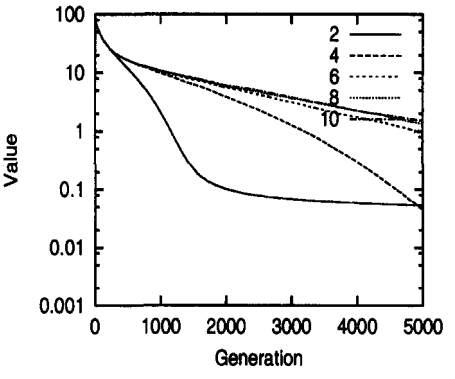



(c) REP.

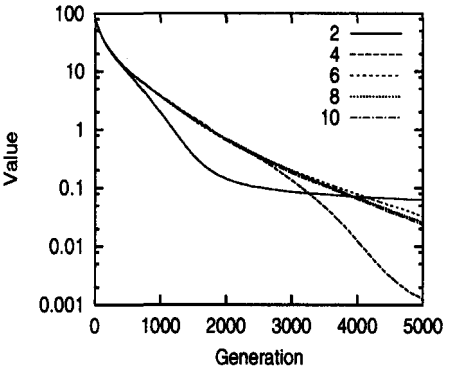
Fig. 4.5: Averaged best results on $f_3$ when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.




(a) CEP.

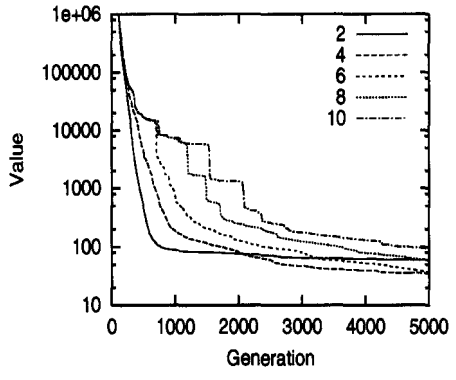



(b) FEP.

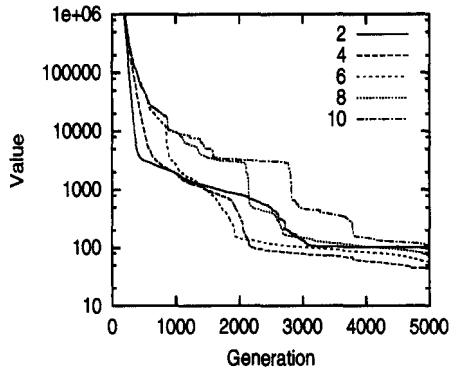



(c) REP.

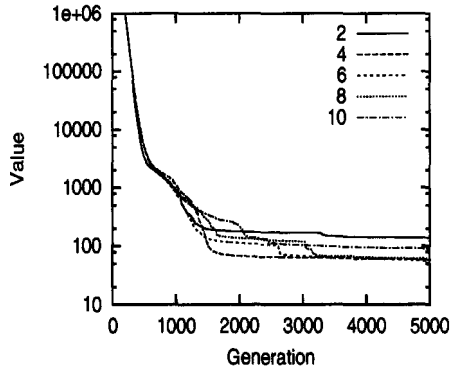
Fig. 4.6: Averaged best results on $f_4$ when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.




(a) CEP.

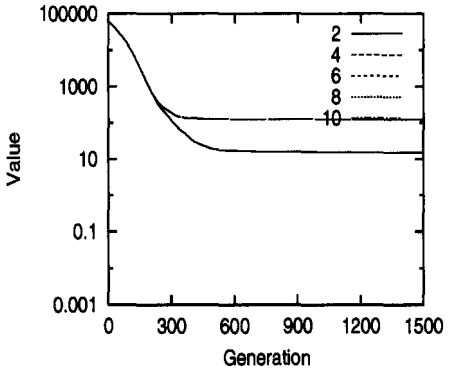



(b) FEP.

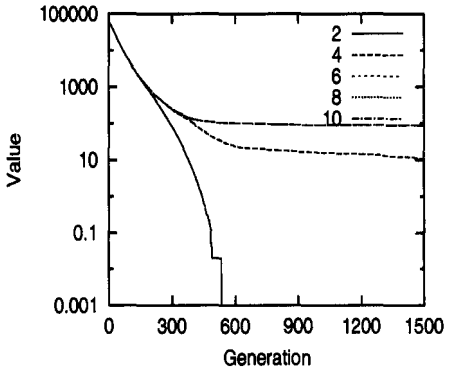



(c) REP.

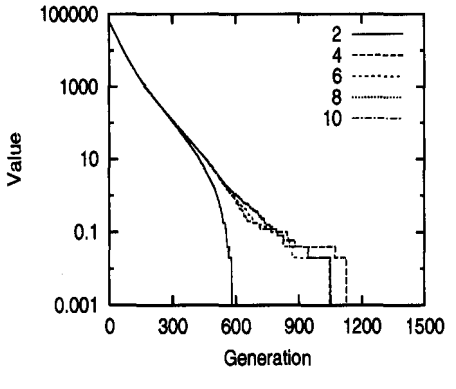
Fig. 4.7: Averaged best results on $f_5$ when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.




(a) CEP.

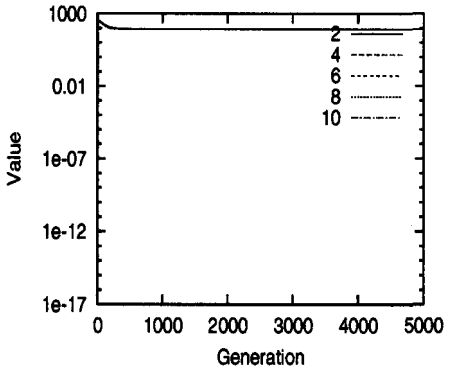



(b) FEP.

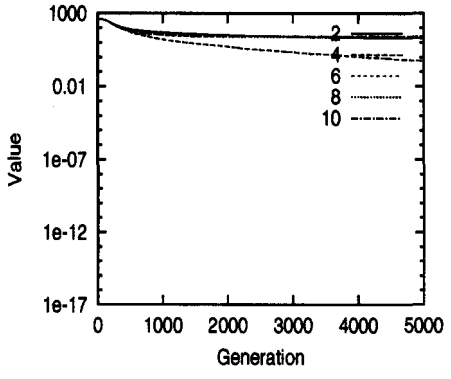



(c) REP.

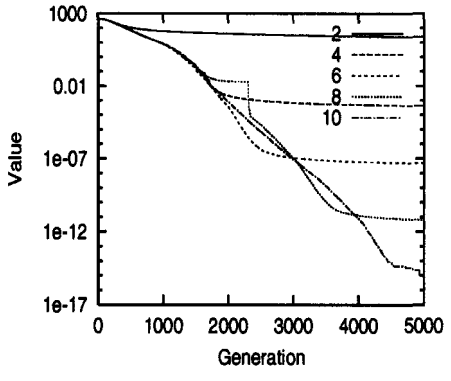
Fig. 4.8: Averaged best results on $f_6$ when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.




(a) CEP.

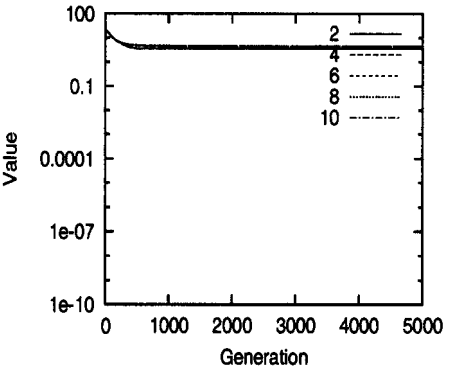



(b) FEP.

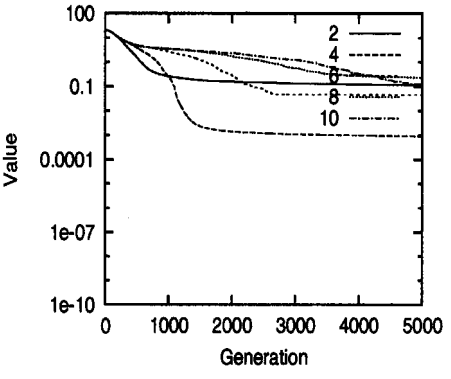



(c) REP.

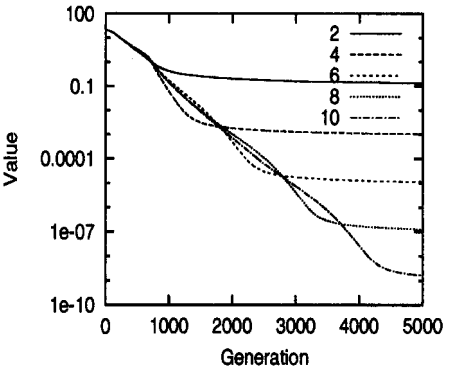
Fig. 4.9: Averaged best results on $f_7$ when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.




(a) CEP.

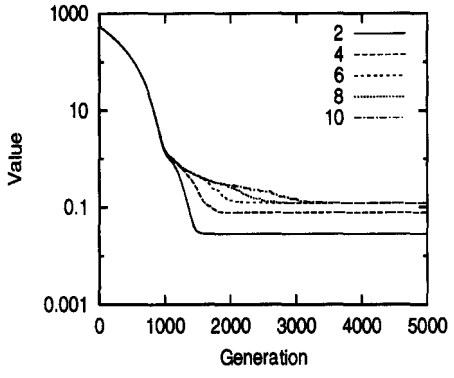



(b) FEP.

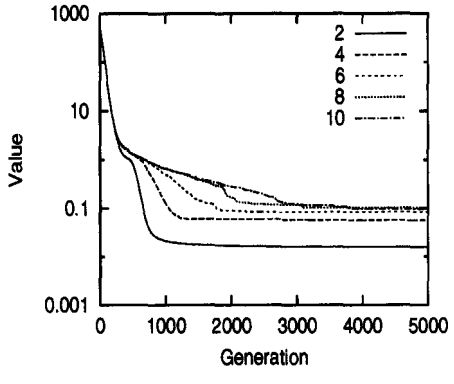



(c) REP.

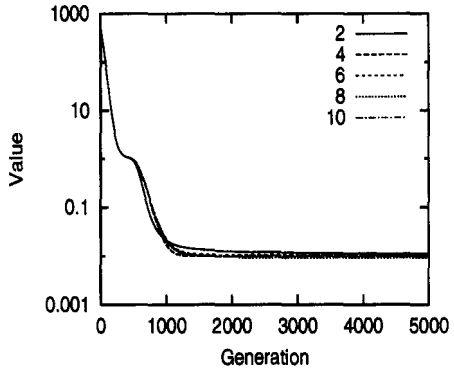
Fig. 4.10: Averaged best results on $f_8$ when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.




(a) CEP.

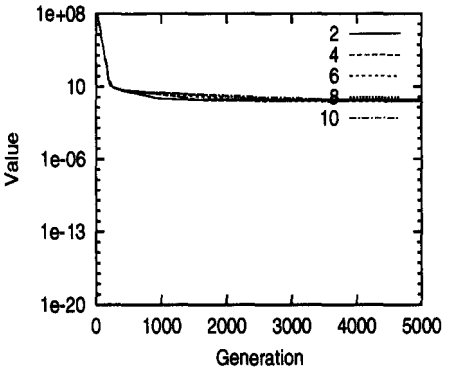



(b) FEP.

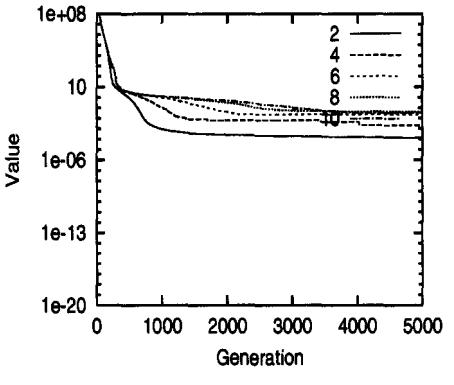



(c) REP.

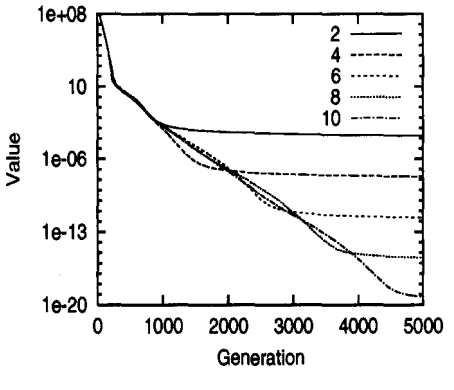
Fig. 4.11: Averaged best results on $f_9$ when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.




(a) CEP.

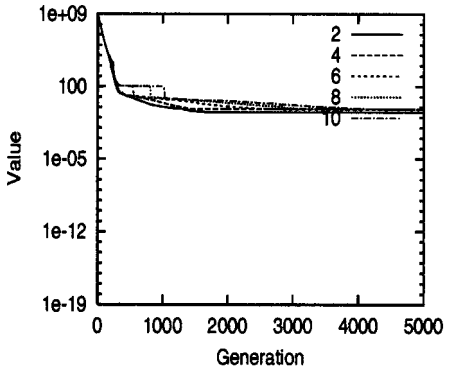



(b) FEP.

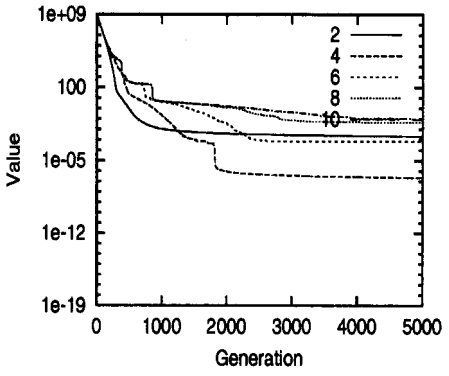



(c) REP.

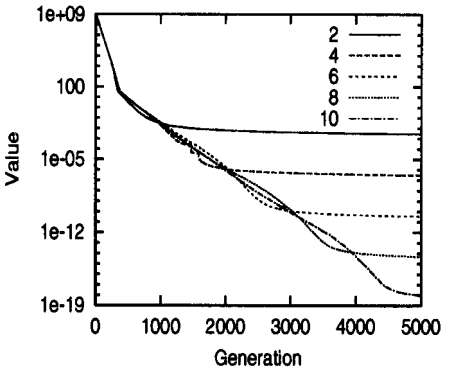
Fig. 4.12: Averaged best results on $f_{10}$ when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.




(a) CEP.




(b) FEP.




(c) REP.

Fig. 4.13: Averaged best results on $f_{11}$ when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.

### 4.2.3 進化ダイナミクス

従来のECでは，最終的に得られた結果のみに焦点をあてた研究が数多くなされてきた．しかしながら，最終結果のみならず統計量に基づく進化ダイナミクスの特徴にも注目しなければならない [16][17][32]．これは，進化ダイナミクスに着目すること，つまり集団がどのように探索し進化したのかを知ることによって，今後の進化アルゴリズムをある問題に適用する際の手掛りになると考えられるからである．そのため，理論面から進化ダイナミクスの解析や [21][22][23][24][33][60][97][123]，進化ダイナミクスの可視化がなされている [95][113]．

最適化手法としてECを用いる際，探索点集団中の適切な多様性を保つことは非常に重要である．EPでは，その世代の親集団と子集団の和から次世代集団を生成するが，集団の探索様式，すなわち進化的振る舞いを調べたい時には，新しく生成された探索点である子集団に関して観測すれば良い．特にここでは，最も基本的関数である $f_1$ を用いて，子集団のいくつかの統計量を観測し，各EPの進化的振る舞いの特徴の違いを把握する．なお，これらの特徴がより明白に現れるようにするために，$\epsilon = 0$ として以下の実験を行った．

#### • 関数値

Fig. 4.14(a)(b)(c) に子集団中のすべての個体の関数値 $f_1(\boldsymbol{x})$ を点でプロットしたものを示す．CEPでは，子集団中のほとんどの個体がよく似た関数値を取っている．FEPでは，そのばらつきはCEPよりは大きいが，最適解への方向が見つけられないことがしばらく続くと，CEPと同じ傾向となる．また，CEP，FEPともに，最適解への接近の仕方が断続的で，最終世代までに十分な解の精度が得られない．一方，REPでは，最適解への接近が継続的に行われ，最終的に $10^{-19}$ まで小さくなる．また，各世代での関数値の分布にも広がりがあり，様々な関数値をとる子孫を常々生成していることが分かる．

#### • 変数値・戦略パラメータ

実数値変数および戦略パラメータの推移を見る．$f_1$ は超球関数であるから $x_i(j)$，$\eta_i(j)$ はすべての $j$ に対して対称である．それゆえ，$x_i(0)$，$\eta_j(0)$(REPにおいては $\eta_{i0}(0)$) の結果を観測することとし，それらをFig. 4.15(a)(b)(c)，Fig. 4.16(a)(b)(c) に示す．

Fig. 4.15の一連の図から分かるように，CEP，FEPではかなり早い時期に子集団すべての $x_i(0)$ が0近傍に集中するようになり，それ以外の点はその後生成されなくなる．REPにおいては，計算世代がかなり進んでも，0から遠い値を取る個体も生成する傾向を示すことが分かる．

同様に，Fig. 4.16の一連の図から，CEP，FEPでは，子集団全体で $\eta_i(0)$ が次第に小さくなっていく．これは，前述の $x_i(0)$ の分布が覆う範囲が次第に狭くなっていくこ

とを裏付けている．一方，REPにおいては，$\eta_{i0}(0)$が世代の経過とともに非常に小さくなるものが多いが，それだけでなく，$10^6$程度の大きさの差をそれぞれ持たせて第2グループ，第3グループ，$\cdots$が存在する．各々がその戦略パラメータサイズを探索戦略とすることから，集団が複数の戦略を同時に使用していることが分かる．

さらに，Fig. 4.17，Fig. 4.18の一連の図に示した$x_i(0)$，$\eta_i(0)$の標準偏差の推移はそれを裏付けている．すなわち，CEP，FEPともに標準偏差が小幅な振れはあるものの基本的に小さくなっていくが，REPでは各世代ごとの振れが次第に大きくなっていく．

なお，ここには示さないが，他の29の変数に関しても同様の特徴が現れることを確認している．

### • Hotellingの$T^2$

集団内の広がり具合をはかる統計量の1つにHotellingの$T^2$がある．これは多次元データ集団の中心から各データ点までの多変量距離を表す尺度であり，この値が大きい個体はデータ群の中心から遠いところに位置し，逆にこの値が小さい個体はデータ群の中心に位置する．それゆえ，大域探索・局所探索のバランスをはかることができる．このHotellingの$T^2$は，以下のように計算される：

$$\boldsymbol{T}^2 = \mu(\mu-1)(\bar{\boldsymbol{x}}-\boldsymbol{\delta})'\boldsymbol{S}^{-1}(\bar{\boldsymbol{x}}-\boldsymbol{\delta}) \tag{4.1}$$

$$\bar{\boldsymbol{x}} = \sum_{i=1}^{\mu}\frac{\boldsymbol{x}_i}{\mu} \tag{4.2}$$

$$\boldsymbol{S} = \sum_{i=1}^{\mu}(\boldsymbol{x}_i-\bar{\boldsymbol{x}})(\boldsymbol{x}_i-\bar{\boldsymbol{x}})' \tag{4.3}$$

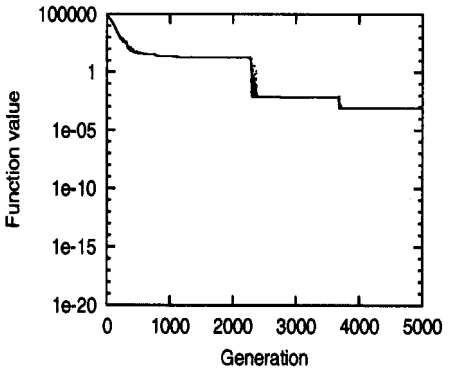
ここで，$\mu$を個体数とし，$\boldsymbol{x}_i$は平均ベクトル$\boldsymbol{\delta}$と共分散行列$\boldsymbol{\sigma_\zeta}$を持つ$n$次元正規分布$N(\boldsymbol{\delta},\boldsymbol{\sigma_\zeta})$に従うものとする．なお，本実験においてこれは成り立たないが，近似的に仮定できるとして用いる．ただし，$\boldsymbol{\delta}$は$\boldsymbol{x}_i$が以下の密度関数$f(\boldsymbol{x})$を持つ時，その期待値として計算される：

$$\begin{aligned} f(\boldsymbol{x}) &= (2\pi)^{-n/2}|\boldsymbol{\sigma_\zeta}|^{-1/2} \\ &\quad \times exp[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\delta})'\boldsymbol{\sigma_\zeta}^{-1}(\boldsymbol{x}-\boldsymbol{\delta})] \qquad (4.4) \\ E(\boldsymbol{x}_i) &= \boldsymbol{\delta} \qquad (4.5) \end{aligned}$$

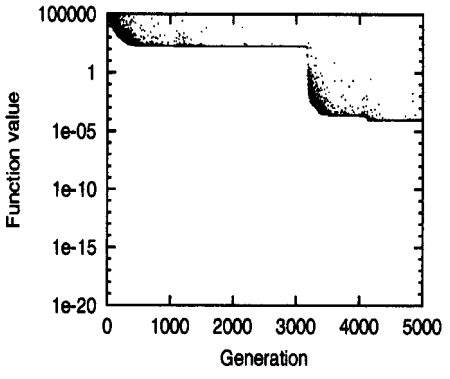
Fig. 4.19(a)(b)(c)に，各EPにおける子集団のHotellingの$T^2$のスナップショットを示す．各図の横軸には子孫100個体をHotellingの$T^2$をもとに降順に並べている．CEPの結果を示したFig. 4.19(a)から，CEPでは，Hotellingの$T^2$は各スナップショットにおいておよそ10から60の間で分布していることが確認できる．また，この分布様式は世代の進行に伴う変化は観察されない．FEPはFig. 4.19(b)にある．CEPよりは

広い範囲に分散してはいるが，世代を経ても集団の分布様式にはあまり大きな違いが現れない．一方，REP の分布様式を示した Fig. 4.19(c) からは，明らかに CEP，FEP のそれらとは違う傾向が見てとれる．REP では，計算初期の分布は FEP に似ているが，次第に 95 以上の大きい値をとるものと 5 以下のごく小さい値をとるものとの割合が増加していく．すなわち，集団の分布において，相対的に広く探索する個体群と局所的に探索する個体群におおまかに分かれることが分かる．

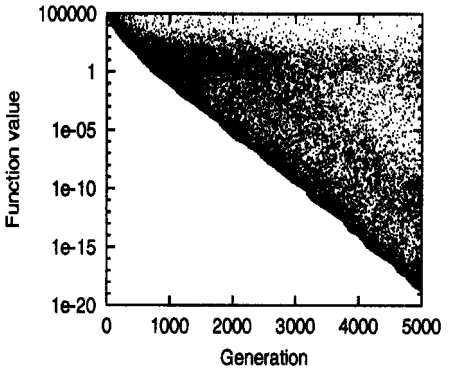
以上より，CEP，FEP では探索初期からその探索戦略を基本的に変えないが，REP ではその世代での実質的探索域に対し広く探索する個体群と中心部分を重点的に探索する個体群に分かれていくことが読み取れる．




(a) CEP.

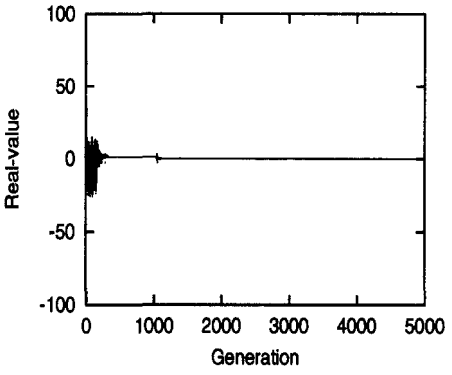



(b) FEP.

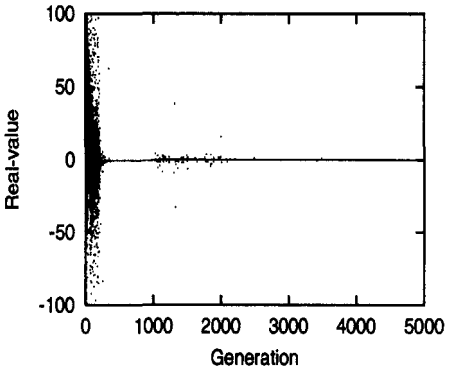



(c) REP.

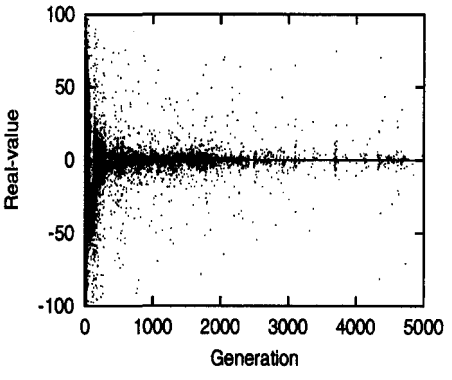
Fig. 4.14: All offspring $[f(\boldsymbol{x})]$ for $f_1$ when $\epsilon = 0$.




(a) CEP.

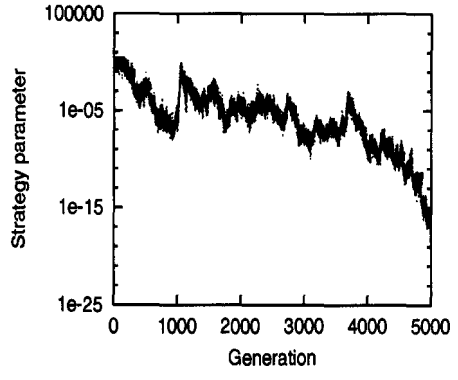



(b) FEP.

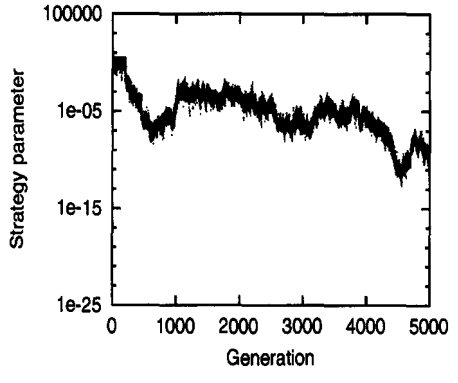



(c) REP.

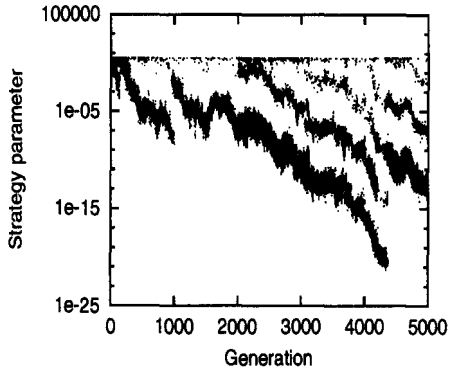
Fig. 4.15: All offspring $[x_i(0)]$ for $f_1$ when $\epsilon = 0$.




(a) CEP.

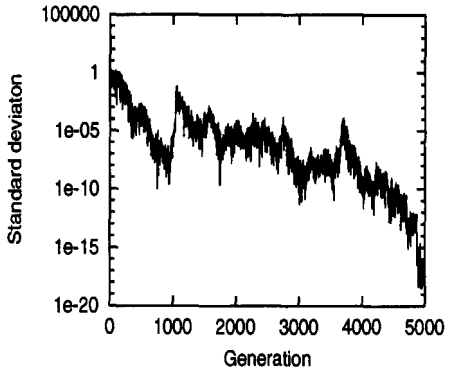



(b) FEP.

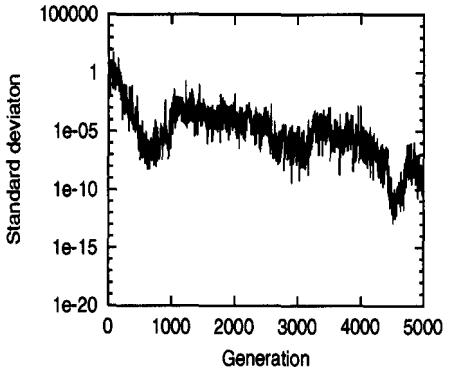



(c) REP.

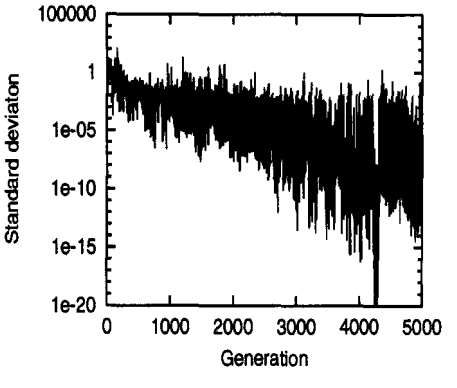
Fig. 4.16: All offspring $[\eta_i(0)]$ for $f_1$ when $\epsilon = 0$.




(a) CEP.

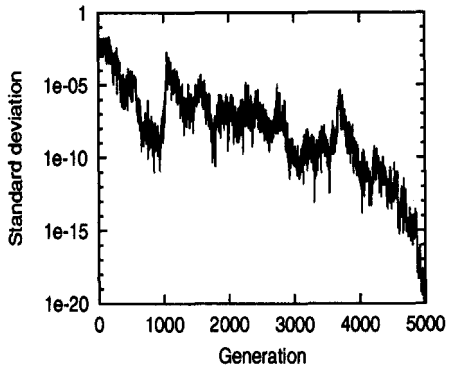



(b) FEP.

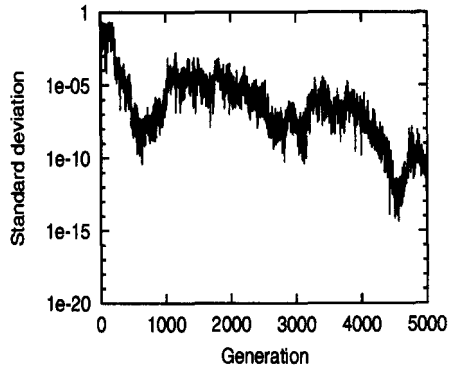



(c) REP.

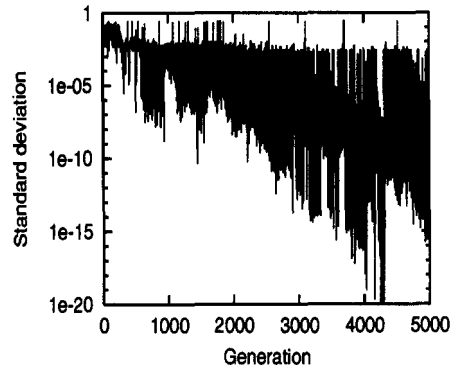
Fig. 4.17: Standard deviation $[x_i(0)]$ for $f_1$ when $\epsilon = 0$.




(a) CEP.

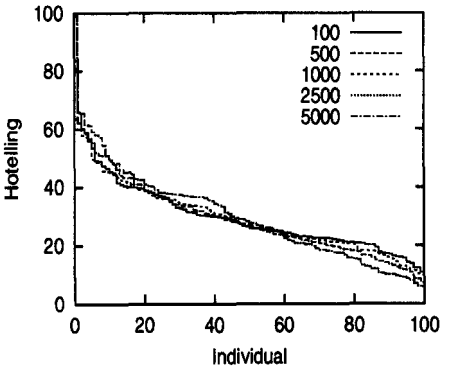



(b) FEP.

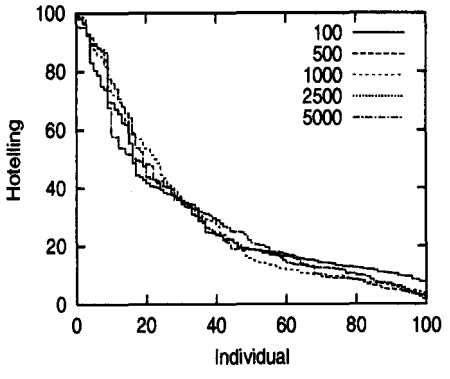



(c) REP.

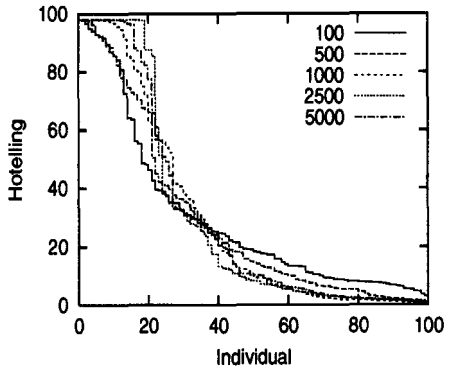
Fig. 4.18: Standard deviation $[\eta_i(0)]$ for $f_1$ when $\epsilon = 0$.




(a) CEP.




(b) FEP.




(c) REP.

Fig. 4.19: The Hotelling's $T^2$ for $f_1$ when $\epsilon = 0$.

### 4.2.4 探索特徴 (まとめ)

以上より，CEP，FEP，REP の探索特徴を以下のようにまとめることができる．

- 単峰性，多峰性関数に関係なく CEP，FEP の探索速度は下限の大きさに依存するが，REP の探索速度は下限の大きさに依存しない．
- CEP，FEP の戦略パラメータは徐々に小さくなり多様性を失うため早期収束の状態に陥りやすくなる．しかし，REP の戦略パラメータは適切なレベルの多様性を維持できるため，進化の方向を見失うことなく最適解に近づくことができる．
- CEP，FEP は，探索戦略を計算世代によって変えないが，REP では集団を実質的探索域で相対的に広く探索するものと中心部分を重点的に探索するものとに次第に分化していく．

## 4.3 ノイズを含むテスト関数を用いた計算機実験

ECを工学的実問題に適用していくことを目指した場合，従来のベンチマーク問題を解く際に仮想的な外乱を与え，評価を下すことは重要であり [1][6][25][27][86][109]，外乱として [12][58] にならった標準正規分布に従うノイズを導入する．

### • テスト関数と諸設定

目的変数以外に含まれるノイズを想定したテスト関数に関する性能比較実験を行う．[12][58] にならい，平均0，標準偏差 $\sigma_\delta$ の標準正規分布に従うノイズ $N(0,\sigma_\delta)$ を加え，以下のようにテスト関数を定義する：

$$F(\boldsymbol{x}) = f(\boldsymbol{x}) + N(0,\sigma_\delta) \tag{4.6}$$

ここで，$\sigma_\delta$ をノイズレベルと呼び，$f(\boldsymbol{x})$ として，Table.4.2 に示す11のテスト関数を用いて計算機実験を行った．$f_1$ から $f_6$ までは単峰性，$f_7$ から $f_{11}$ は多峰性関数である．すべてのテスト関数は30次元の探索空間を持ち，ノイズが存在しない場合の大域最小値は0である．

CEP と FEP および REP はともに，[116][118][120][121] に従って，$\mu = 100$ として計算する．ノイズレベルを $\sigma_\delta$={0.0，0.001，0.01，0.1，1.0} の値に変えて，それぞれ50回独立に試行を繰り返した．戦略パラメータの上限 $\eta_{max}$ は探索範囲が比較的狭い $f_7$ のみ1.0，その他では3.0とし，$m = 5$ とし，$g_{dup}$，$g_{del}$，$g_{inv}$ の適用確率 (P) を，それぞれ，$P(g_{dup}) = 0.6$，$P(g_{del}) = 0.3$，$P(g_{inv}) = 0.1$ とした．なお，CEP と FEP の実験は，REP で CEP と FEP に等価になるパラメータ値，すなわち，$P(g_{dup}) = 1$，$P(g_{del}) = 0$，$P(g_{inv}) = 0$ として行った [78]．

REP におけるこれらの推奨パラメータ値は，予備的な実験結果から決定した．また，同実験から，パラメータ値の変化に関して REP は非常に頑健であることが分かったため，パラメータ値チューニングにはそれほど注意を払っていない．それゆえ，詳細な実験を通してこれらを決定すれば，パフォーマンスは更に向上すると思われる．

### • 結果 (単峰性関数)

単峰性関数 $f_1$ から $f_6$ に関して，計算結果を Fig. 4.20 から Fig. 4.25 に示す．それぞれ，縦軸には50回の試行によって得られた親集団の最良値を平均した数値をとり，横軸には世代数をとったものである．

$f_1$ に関して，CEP と FEP では300世代以降ノイズによって探索速度が遅くなるのに対し，REP ではノイズの影響は無く，全てのノイズレベルで高速に負の領域まで探索している．同様に，$f_2$ では，CEP と FEP に対し REP はノイズの影響は無く高速に探索している．$f_3$ では，CEP，FEP，REP ともノイズの影響がほとんど見られないが，$f_4$ と $f_5$ では，CEP，FEP，REP ともノイズの影響を受けている．特に，探索性能

Table. 4.4: Summary of experiments for noisy test functions.

| | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ | $f_8$ | $f_9$ | $f_{10}$ | $f_{11}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| CEP | × | × | ○ | × | × | × | × | × | × | × | × |
| FEP | × | × | ○ | × | × | × | × | × | × | × | × |
| REP | ○ | ○ | ○ | × | × | ○ | ○ | × | × | ○ | ○ |

の差は $f_6$ で顕著に見られ，CEP と FEP が大域最小値に近づけないことに対し，REP は全てのノイズレベルで負の領域まで探索している．

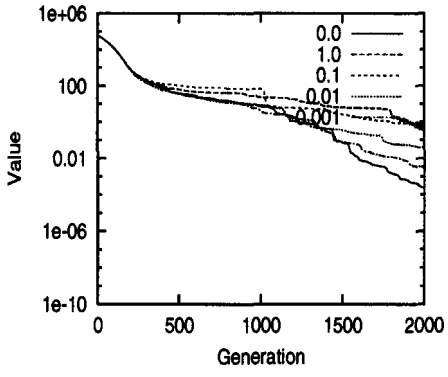
## • 結果 (多峰性関数)

多峰性関数 $f_7$ から $f_{11}$ に関して，計算結果を Fig. 4.26 から Fig. 4.30 に示す．それぞれ，縦軸に 50 回の試行によって得られた親集団の最良値を平均した数値をとり，横軸に世代数をとったものである．

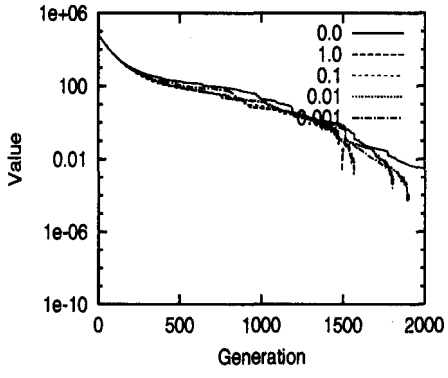
$f_7$ に関して，CEP は 300 世代以降収束し，FEP では局所解から抜け出せていない．一方，REP では全てのノイズレベルで負の領域まで探索している．$f_8$ と $f_9$ では CEP，FEP，REP ともノイズの影響を受けるが，$f_{10}$ と $f_{11}$ では，CEP と FEP に対し REP はノイズの影響を受けず，高速に局所解に陥ることなく探索している．

## • 計算機実験結果 (まとめ)

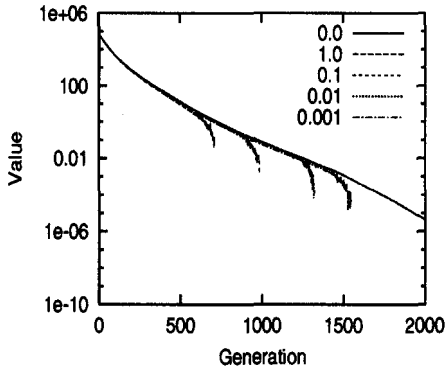
計算機実験結果によって得られたノイズに対する頑健性を Table.4.4 にまとめる．CEP と FEP は，$f_3$ 以外の全てのテスト関数においてノイズの影響を受けた．一方，REP は $f_1$，$f_2$，$f_3$，$f_6$，$f_7$，$f_{10}$，$f_{11}$ ではノイズに関して頑健であった．




(a) CEP.

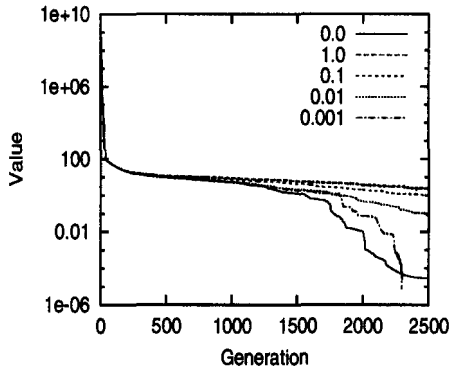



(b) FEP.

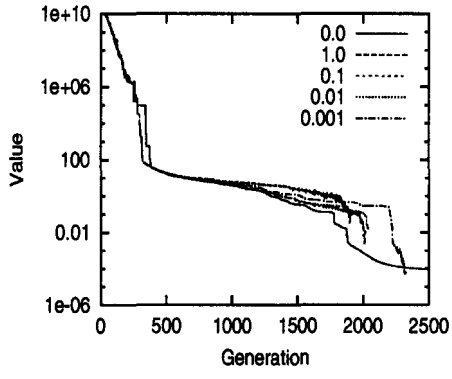



(c) REP.

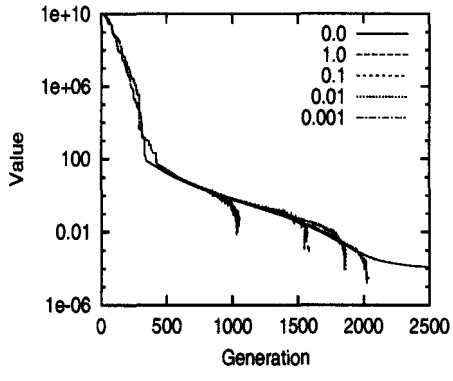
Fig. 4.20: Averaged best results on $f_1$ with output noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.




(a) CEP.

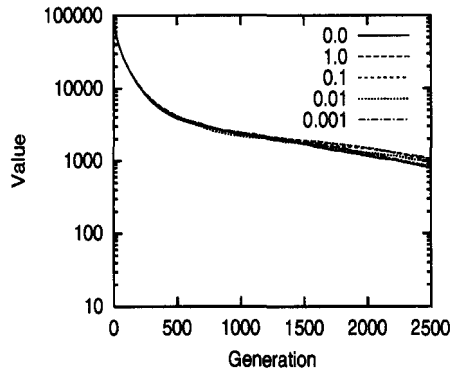



(b) FEP.

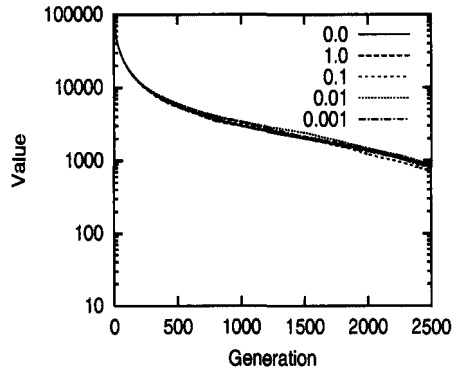



(c) REP.

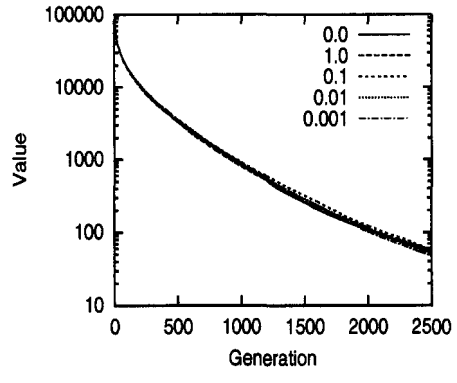
Fig. 4.21: Averaged best results on $f_2$ with output noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.




(a) CEP.

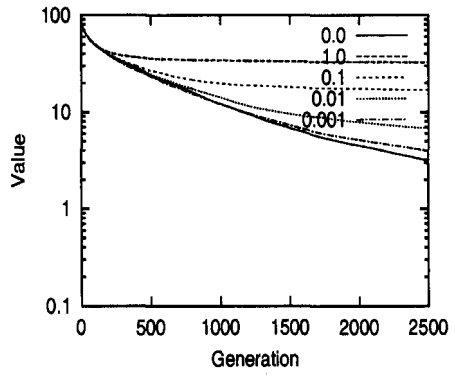



(b) FEP.

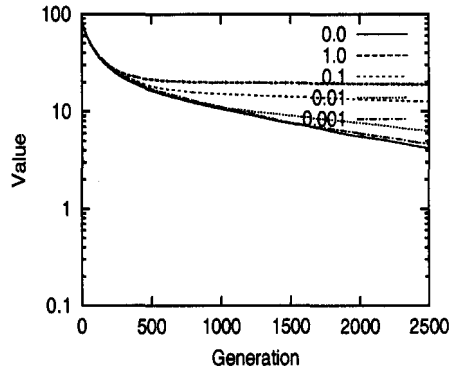



(c) REP.

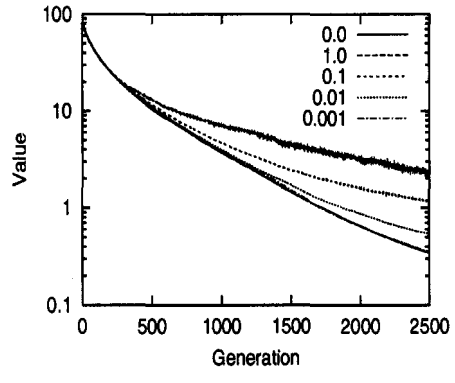
Fig. 4.22: Averaged best results on $f_3$ with output noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.




(a) CEP.

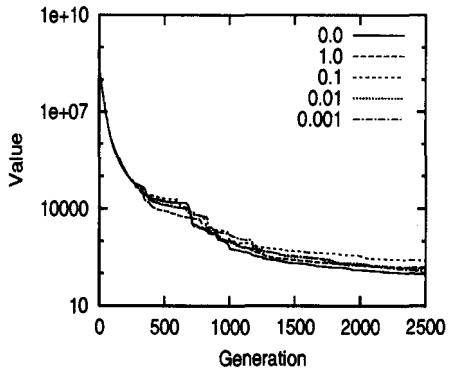



(b) FEP.

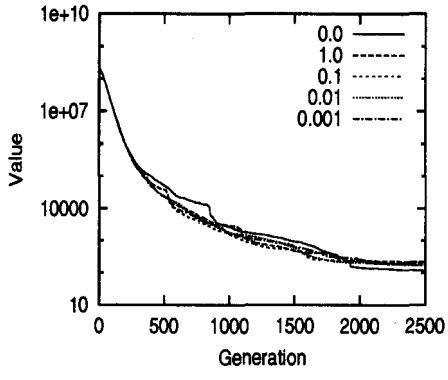



(c) REP.

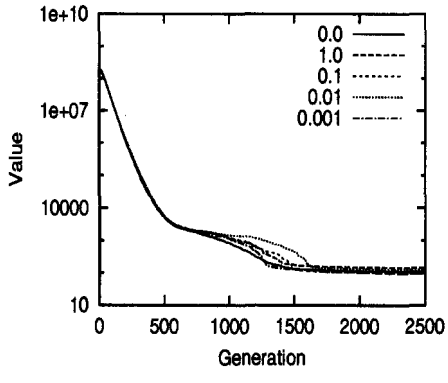
Fig. 4.23: Averaged best results on $f_4$ with output noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.




(a) CEP.

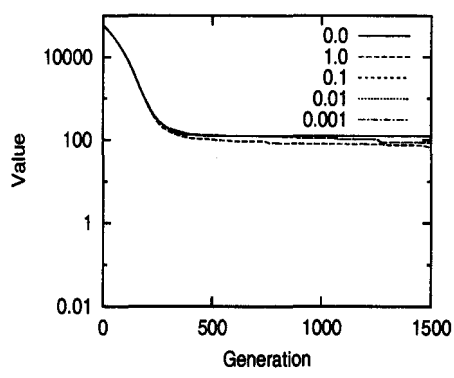



(b) FEP.

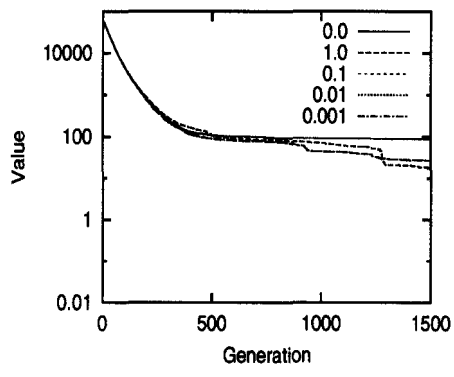



(c) REP.

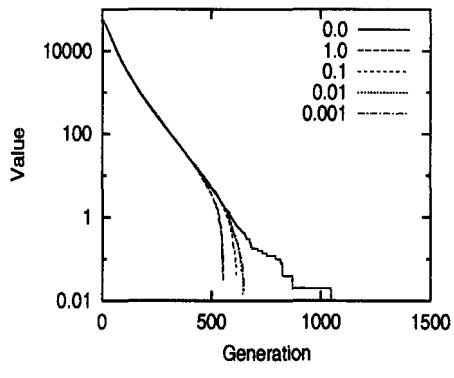
Fig. 4.24: Averaged best results on $f_5$ with output noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.




(a) CEP.

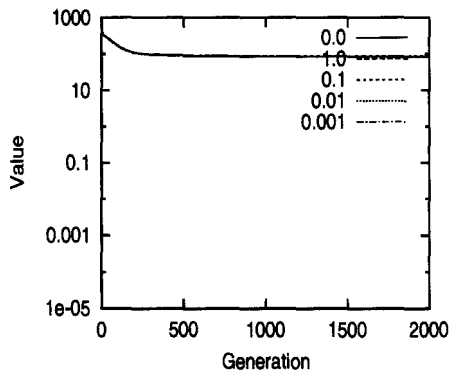



(b) FEP.

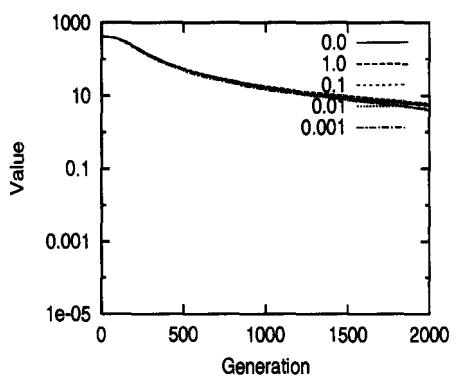



(c) REP.

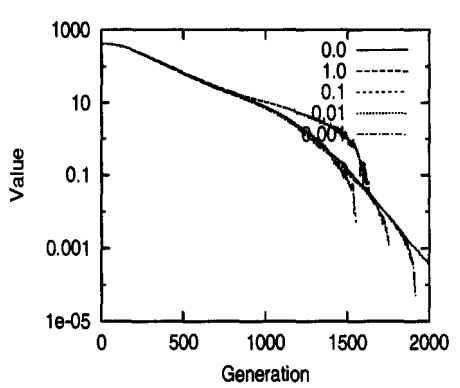
Fig. 4.25: Averaged best results on $f_6$ with output noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.




(a) CEP.

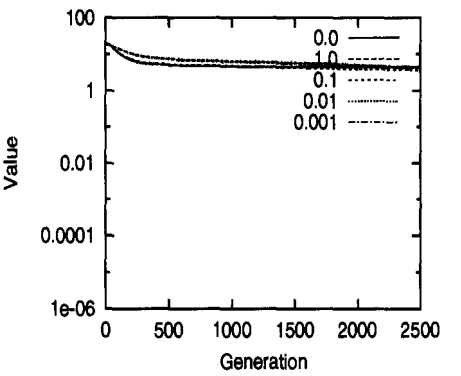



(b) FEP.

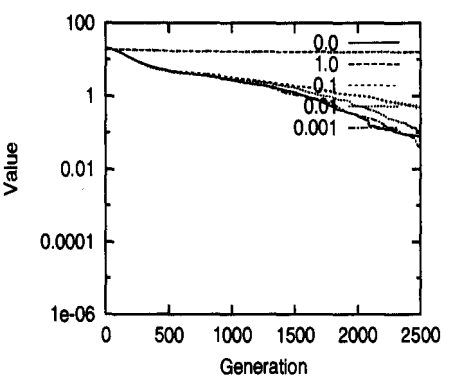



(c) REP.

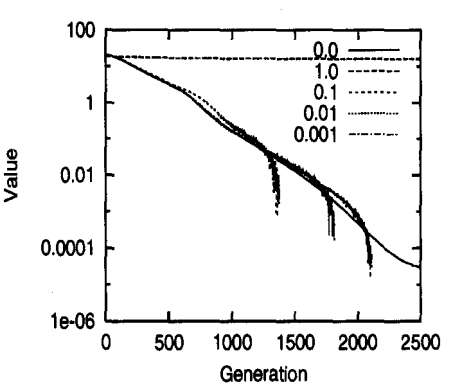
Fig. 4.26: Averaged best results on $f_7$ with output noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.




(a) CEP.

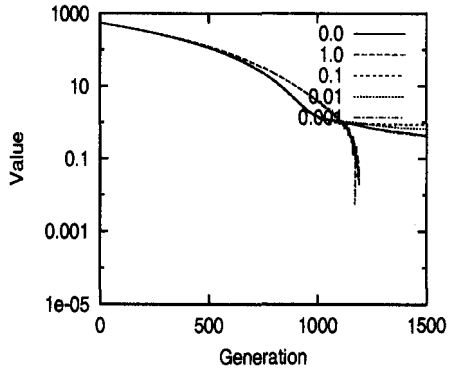



(b) FEP.

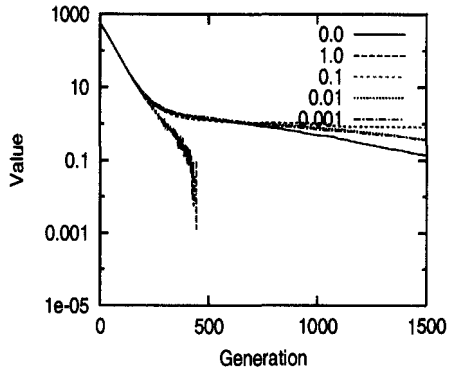



(c) REP.

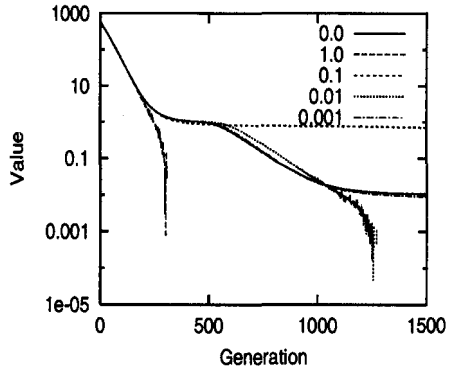
Fig. 4.27: Averaged best results on $f_8$ with output noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.




(a) CEP.

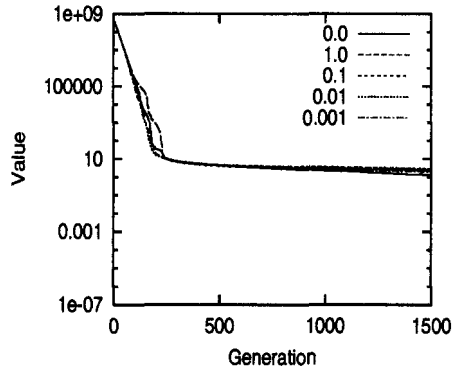



(b) FEP.

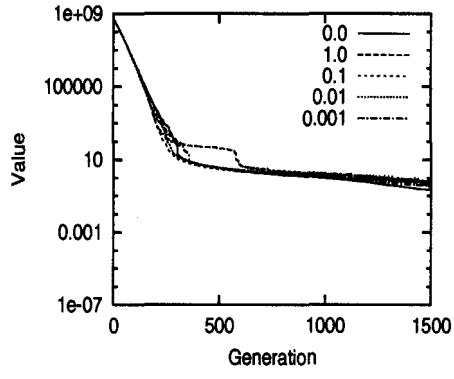



(c) REP.

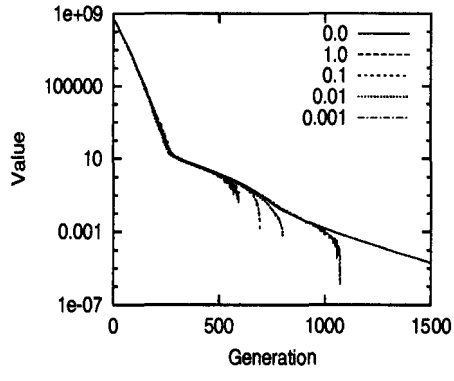
Fig. 4.28: Averaged best results on $f_9$ with output noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.




(a) CEP.

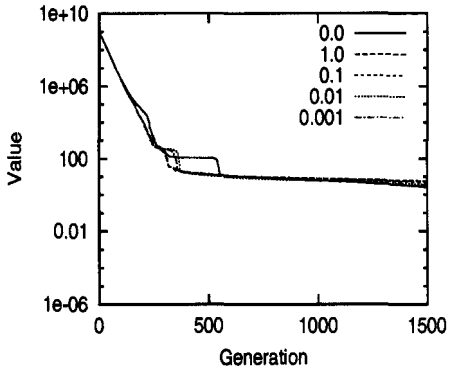



(b) FEP.

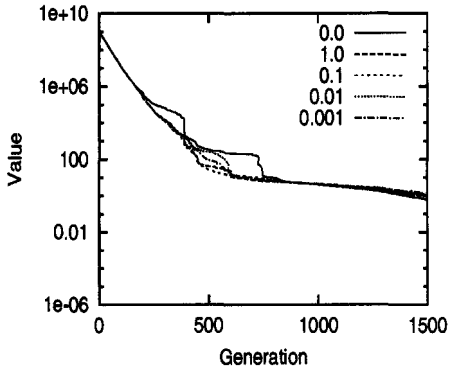



(c) REP.

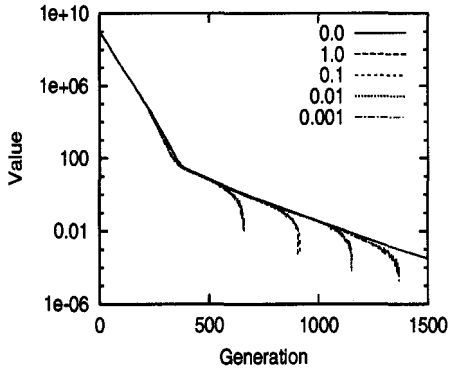
Fig. 4.29: Averaged best results on $f_{10}$ with output noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.




(a) CEP.




(b) FEP.




(c) REP.

Fig. 4.30: Averaged best results on $f_{11}$ with output noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.

## 4.4 多指ハンドによる安定把持計画問題への応用

工学的実問題への応用として，非線形制約条件付最適化問題を取り上げる．特に，多指ハンドにおける指先力計画・指先位置計画を同時に決定する最適化問題を取り扱う．この多指ハンドの問題において，把持する物体のつり合い方程式のモーメントのつり合いは力と位置の積を含むため，指先位置を可変とすると力と位置両方が変数となりつり合い方程式は非線形になる．さらに，それを満たす位置と力の集合は凸でなく，このような制約条件を満たす指先位置の最適化は取り扱いが極めて難しくなる．特に，多指ロボットハンド問題で安定性を考慮すると，評価関数はますます複雑になり，最小二乗法等では解くことが困難となる．そこで，進化的プログラミングの評価関数にペナルティ関数を導入することによって非線形制約条件を取り扱い，非線形の力学的拘束条件を満たした指先力計画と指先位置計画を行う．本節では，計算機実験によって従来手法 (CEP) と提案手法 (REP) の性能を比較検証することを目的とする．

### 4.4.1 平衡条件と摩擦条件

$n$ 本の指による把持の場合，平衡方程式は以下のように表される．

$$\sum_{i=1}^{n} \boldsymbol{f}_i = -m\boldsymbol{g} \tag{4.7}$$

$$\sum_{i=1}^{n} \boldsymbol{\rho}_i \times \boldsymbol{f}_i = 0 \tag{4.8}$$

ここで，$\boldsymbol{f}_i$ は $i$ 番目の指の力ベクトル，$m$ は把持物体の質量であり，$\boldsymbol{g}$ は重力加速度である．また，$\boldsymbol{\rho}_i$ は $i$ 番目の指の重心からの位置ベクトルである．

また指先と物体との摩擦条件は

$$|\, \boldsymbol{n}_i \times \boldsymbol{f}_i \,| \leq \mu(\boldsymbol{f}_i \cdot \boldsymbol{n}_i) \;\; i = 1, \ldots, n \tag{4.9}$$

(但し，$\boldsymbol{n}_i$ は物体表面の法線ベクトル，$\mu$ は摩擦係数）で表わされる．本論文では指のコンフィグレーションは考えないが，指先力に制限 $f_{max}$ を設け，指先力の拘束を以下のように定義する．

$$|\, f_{ix} \,| \leq f_{max} \tag{4.10}$$

$$|\, f_{iy} \,| \leq f_{max} \tag{4.11}$$

$$0 < f_{iz} \leq f_{max} \tag{4.12}$$

ここで，$f_{ix}, f_{iy}, f_{iz}$ は $i$ 番目の指の $x, y, z$ 方向の指先力である．

Coulomb の摩擦法則より，摩擦条件は以下のように表される．

$$\sqrt{f_{ix}^2 + f_{iy}^2} \leq \mu_0 f_{iz} \tag{4.13}$$

ここで，$\mu_0$ は最大静止摩擦係数である．

### 4.4.2 Liapunovの安定条件と接触安定条件

Liapunovの安定条件は次のように表される.

$$L \geq 0, \qquad L^2 + S \geq 0, \qquad LS - V \geq 0 \tag{4.14}$$

ここで,

$$L = \sum_{i=1}^{n} \boldsymbol{\rho}_i \cdot \boldsymbol{f}_i, \tag{4.15}$$

$$S = \sum_{i=1}^{n} \sum_{j=i+1}^{n} (\boldsymbol{\rho}_i \times \boldsymbol{\rho}_j) \cdot (\boldsymbol{f}_i \times \boldsymbol{f}_j), \tag{4.16}$$

$$V = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \sum_{k=j+1}^{n} \{\boldsymbol{\rho}_i \cdot (\boldsymbol{\rho}_j \times \boldsymbol{\rho}_k)\} \cdot \{\boldsymbol{f}_i \cdot (\boldsymbol{f}_j \times \boldsymbol{f}_k)\}. \tag{4.17}$$

式(4.14)は必ずしも片側拘束条件下の把持計画で満たされるとは限らない．そこで各指の先端剛性を制御して把持対象物体に関する安定性を実現する．これは仮想線形バネに相当し，次のように表される.

$$\Delta \boldsymbol{f}_i = -k_{pi} \Delta \boldsymbol{x}_i, \quad i = 1, \ldots, n \tag{4.18}$$

ここで，$\Delta \boldsymbol{x}_i = \Delta \boldsymbol{x} + \boldsymbol{\theta} \times \boldsymbol{\rho}_i$ であり，$\Delta \boldsymbol{x}$ は物体座標系における直交成分の変位であり，$\boldsymbol{\theta}$は回転成分の変位である.

制御則(4.18)の下で安定条件は式(4.14)より次のようになる.

$$L' \geq 0, \qquad L'^2 + S' \geq 0, \qquad L'S' - V' \geq 0 \tag{4.19}$$

ここで,

$$L' = \sum_{i=1}^{n} \boldsymbol{\rho}_i \cdot \Delta \boldsymbol{f}_i, \tag{4.20}$$

$$S' = \sum_{i=1}^{n} \sum_{j=i+1}^{n} (\boldsymbol{\rho}_i \times \boldsymbol{\rho}_j) \cdot (\Delta \boldsymbol{f}_i \times \Delta \boldsymbol{f}_j), \tag{4.21}$$

$$V' = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \sum_{k=j+1}^{n} \{\boldsymbol{\rho}_i \cdot (\boldsymbol{\rho}_j \times \boldsymbol{\rho}_k)\} \cdot \{\Delta \boldsymbol{f}_i \cdot (\Delta \boldsymbol{f}_j \times \Delta \boldsymbol{f}_k)\}. \tag{4.22}$$

同様に，摩擦条件は次のようになり，これを接触安定条件と定義する.

$$| \boldsymbol{n}_i \times (\boldsymbol{f}_i + \Delta \boldsymbol{f}_i) | \leq \mu \{(\boldsymbol{f}_i + \Delta \boldsymbol{f}_i) \cdot \boldsymbol{n}_i\} \tag{4.23}$$

### 4.4.3 評価関数

式 (4.18) と (4.23) より,

$$| \Delta \boldsymbol{x}_i | \leq \frac{\mu (\boldsymbol{n}_i \cdot \boldsymbol{f}_i) - | \boldsymbol{n}_i \times \boldsymbol{f}_i |}{k_{pi}(1+\mu)} \tag{4.24}$$

ただし，回転成分は無視できると仮定する．線形成分の変位が最大になることが望ましいため，式 (4.7)(4.8)(4.10)(4.11)(4.12)(4.13)(4.19) を拘束条件とし，以下の評価関数を最大化する最適化問題として定式化する:

$$F = \max\{\min_i \frac{\mu (\boldsymbol{n}_i \cdot \boldsymbol{f}_i) - | \boldsymbol{n}_i \times \boldsymbol{f}_i |}{k_{pi}(1+\mu)}\} \tag{4.25}$$

また各変数は $| \Delta \boldsymbol{f} | \leq \Delta \boldsymbol{f}_{max}$, $| \Delta \boldsymbol{c} | \leq \Delta \boldsymbol{c}_{max}$, $\Delta k_{p_i} \leq \Delta k_{p_{max}}$ なる定義域を持つ．Kim[69] の方法を用いて制約条件をペナルティ関数として導入する．

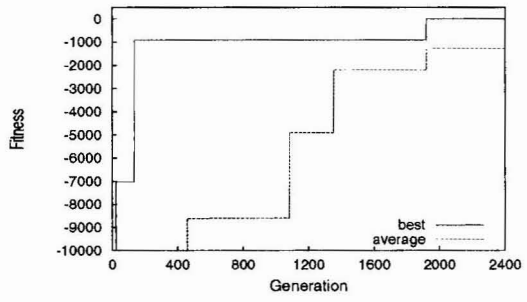
### 4.4.4 計算機実験

3 次元における 3 本指による円柱 ($\mu = 0.3, m = 0.8$) の把持をタスクとする．各遺伝子は，物体表面の位置・内力ベクトル・指先の剛性を変数に持つ．個体数は第一相・第二相とも 40 個体，世代数は第一相を 400，第二相を 2000 世代とし試行回数を 10 回とする．

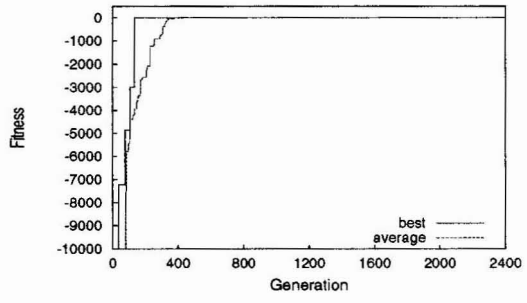
### 4.4.5 実験結果

2400 世代までの各試行の平均と最良結果を Fig. 4.31(a)(b) に示す．$F$ は常に正の値をとるので評価値 $\Phi_1(x)$,$\Phi_2(x)$ が負であることは制約条件を満たさず，ペナルティを与えられていることを表す．従来手法である CEP を用いた場合，10 回の試行中 8 回制約条件を満たす個体が現れた．REP を用いた場合，最終世代において最良結果を示した試行では，ペナルティを与えられる個体は世代が経るにつれて次第に淘汰され，132 世代で全ての制約条件を満たす個体が現れた．また全試行において，制約条件が満たされていることが確認された．Fig. 4.31(a)(b) からわかるように解の精度・収束性ともに REP を用いた場合の方が良い結果を得ている．

最終世代での最良結果の指先位置及び相対指先力を Fig. 4.32 に示す．ほぼ同じ円柱上の高さに接触点を持ち，一組の指ともう一本の指が対向するように配置される結果となっている．

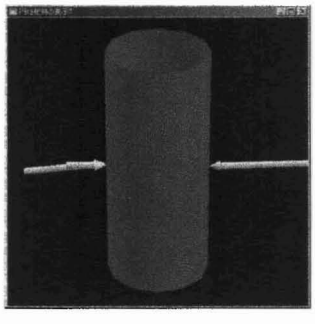



(a) CEP.

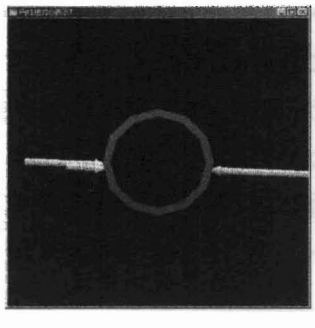



(b) REP.

Fig. 4.31: The averaged best results.



(a) Front.



(b) Top.

Fig. 4.32: Force and position at the last generation.

## 4.5 結言

本章では，進化戦略の一種として考えられる進化的プログラミングに関する自己適応の拡張法について議論している．実関数最適化問題における標準テスト関数を対象とした計算機実験により，従来の進化的プログラミングは，戦略パラメータが極端に小さくなり，早期収束の状態に陥りやすいことを確認した．それに対し，第3章で定式化した提案手法は，標準テスト関数に対し，早期収束の状態を回避でき，戦略パラメータの下限設定に対する頑健性が従来手法に比べて向上した．進化ダイナミクスの観測を行った結果，従来手法に比べて提案手法は局所探索と大域探索のバランスが強化され，探索領域を伸縮しているため，遺伝的浮動の効果が発揮されていると考えられる．また，ノイズを含むテスト関数に関しても，従来手法と比較し，提案手法の有効性を実証した．さらに，工学的実問題への応用として多指ハンドよる物体の把持問題を取り上げ，Liapunovの安定条件及び接触安定条件を満たす指先位置・指先力計画の最適化を行った．この問題は27の非線形制約条件付き最適化問題であり，超多峰性問題の一つと考えられる．特に，進化的プログラミングの特徴である最良個体が生存する点は，全ての拘束条件を一度充足すれば少なくとも1個体が拘束条件を充足したまま生存することが約束されるため，この問題に有効であると考えられる．しかしながら，従来手法を用いた計算機実験では全ての拘束条件を満たす解の発見が困難であることを確認した．それに対し，提案手法では，全ての拘束条件を満たす解を高速に発見することが可能となり，非線形制約条件付き最適化問題に対する提案手法の有用性を実証した．以上より，進化的プログラミングにおける分子進化の中立説に動機付けられた自己適応の拡張法の妥当性を示したと言える．

# 第5章 $(\mu, \lambda)$-ESにおける自己適応の拡張

## 5.1 緒言

本章では，進化戦略 (Evolution Strategies: ES) の $(\mu, \lambda)$-ES における自己適応の拡張法に関して，問題の次元に重点をおいて議論する．本章では，$(\mu, \lambda)$-ES において従来の Gauss 型突然変異を用いる手法を Classical-ES (CES)，Cauchy 型突然変異を用いる手法を Fast-ES (FES)，提案する新しい ES の拡張法を Robust-ES (RES) として参照する．本章の計算機実験では，実関数最適化問題において一般に用いられている標準テスト関数によって従来手法 (CES，FES) と提案手法 (RES) を比較検証することによって，自己適応の問題点に対する提案手法の有効性を明らかにし，進化ダイナミクスの解析により手法の特性を明らかにする．また，工学的問題に応用する上で重要となるノイズに対する頑健性を検証した後に，実問題に応用しその有用性を検証する．特に，工学的実問題への応用として，自律移動ロボットのナビゲーション問題における Continuous-Time Recurrent Neural Networks (CTRNNs) の進化的設計を取り扱う．この CTRNNs は，実数値の入出力を取り扱うことが可能であり，連続時間を対象にしている．そのため，回路規模に対して変数は指数関数的に増大し，標準テスト関数に比べて次元の高い最適化問題となる．

Table. 5.1: Contents of Section 5.

| 5章 | 5.2 | 5.3 | 5.4 |
|---|---|---|---|
| 問題 | 標準テスト関数 | ノイズを含むテスト関数 | CTRNNs の設計問題 |
| 手法 | CES,FES,RES | CES,FES,RES | CES,RES |

## 5.2 標準テスト関数を用いた計算機実験

### 5.2.1 テスト関数と諸設定

提案手法であるRESの有効性と進化過程の特徴を明らかにするために，従来手法としてCESとFESを取り上げ，Table.4.2に示す11のテスト関数を用い計算機実験を行った[89][90][77]．これらは，それぞれ，Hypersphere関数($f_1$)，Schwefelの問題2.22($f_2$)，Schwefelの問題1.2($f_3$)，Schwefelの問題2.21($f_4$)，Rosenbrock関数($f_5$)，Step関数($f_6$)，Rastrigin関数($f_7$)，Ackley関数($f_8$)，Griewank関数($f_9$)，Penalized関数(P8)($f_{10}$)，Penalized関数(P16)($f_{11}$)である．また，$f_1$から$f_6$までは単峰性，$f_7$から$f_{11}$は多峰性関数である．すべてのテスト関数は30次元の探索空間を持ち，大域最小値は0である．

CESとFESおよびRESはともに，[117][119]に従って，$(\mu, \lambda) = (30, 200)$，相関突然変異および組換えを用いないこととした．戦略パラメータの上限$\eta_{max}$は探索範囲が比較的狭い$f_7$のみ1.0，その他では3.0とし，下限$\epsilon$を$10^{-2}$，$10^{-4}$，$10^{-6}$，$10^{-8}$，$10^{-10}$と5つの値に変えて，それぞれ50回試行を繰り返した．さらに，RES [89][90][77]では，$m = 6$として，$g_{dup}$，$g_{del}$，$g_{inv}$の適用確率(P)を，それぞれ，$P(g_{dup}) = 0.6$，$P(g_{del}) = 0.3$，$P(g_{inv}) = 0.1$とした．なお，CESとFESの実験は，RESでCESとFESに等価になるパラメータ値，すなわち，$P(g_{dup}) = 1$，$P(g_{del}) = 0$，$P(g_{inv}) = 0$として行った．

### 5.2.2 基本探索性能

- **計算機実験結果 (単峰性関数)**

単峰性関数$f_1$から$f_6$の計算結果をFig. 5.1からFig. 5.6に示す．それぞれ，縦軸に関数値，横軸に世代をとった．

CESとFESによる$f_1$の結果を示したFig. 5.1(a)とFig. 5.1(b)より，CES，FESともに下限が$10^{-4}$より小さい場合に200世代付近で，早期収束の傾向が顕著に見られ探索速度が落ちている．一方，RESにおいては，Fig. 5.1(c)から分かるように，下限によってそれほど探索速度の差異は無く，小さい下限を用いるほど，精度良く大域最適解に近づくことができている．

そして，これらの傾向は，$f_2$から$f_5$でも見られる．Fig. 5.2より，$f_2$では下限が$10^{-4}$より小さい場合に100世代付近で，Fig. 5.3より$f_3$では下限が$10^{-6}$より小さい場合に200世代付近で，Fig. 5.4より$f_4$では下限が$10^{-6}$より小さい場合に300世代付近で，Fig. 5.5より$f_5$では下限が$10^{-6}$より小さい場合に200世代付近で，早期収束の傾向が顕著に見られ探索速度が落ちている．そのため，下限が小さくなるにしたがってRESの優位性が現れている．

特に，$f_6$の結果を示したFig. 5.6から，$f_1$から$f_5$で見られるCESとFESとRES

Table. 5.2: Summary of experiments for standard test functions.

| | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ | $f_8$ | $f_9$ | $f_{10}$ | $f_{11}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| CES | × | × | × | × | × | × | × | × | × | × | × |
| FES | × | × | × | × | × | × | × | × | × | × | × |
| RES | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |

の探索の推移の違いはさらに明確に現れる．CESは全く最適解0を発見できず，FESは$\epsilon = 10^{-2}$の時には平均175世代で大域最適解に到達しているものの，$10^{-6}$より小さくなってしまうと最適解を発見できていないことが分かる．一方，RESでは，どの値でも200世代前後で最適解に到達していることが分かる．

## • 計算機実験結果 (多峰性関数)

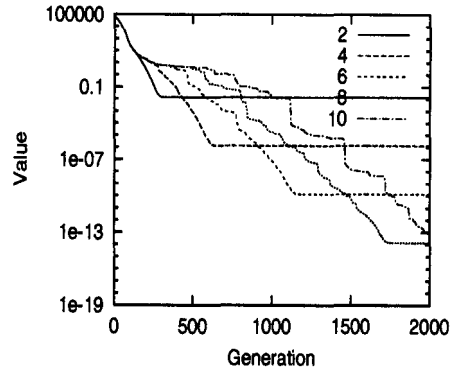
つぎに，多峰性関数$f_7$から$f_{11}$の計算結果をFig. 5.7からFig. 5.11に示す．それぞれ，縦軸に関数値，横軸に世代をとった．

CESとFESによる$f_7$の結果を示したFig. 5.7(a)とFig. 5.7(b)より，CES，FESともに下限が$10^{-4}$より小さい場合に100世代付近で，早期収束の傾向が顕著に見られ探索速度が落ちている．また，FESにおいて下限が$10^{-4}$の場合にやや改善が見られるもののその他ではFES，CESとも明らかに集団が局所解から抜け出せない傾向も示している．一方，RESにおいては，Fig. 5.7(c)から分かるように，単峰性関数の傾向と同様，下限の違いによる探索速度の差異もほとんど無く，小さい下限を用いるほど精度良く大域最適解に近づくことができている．

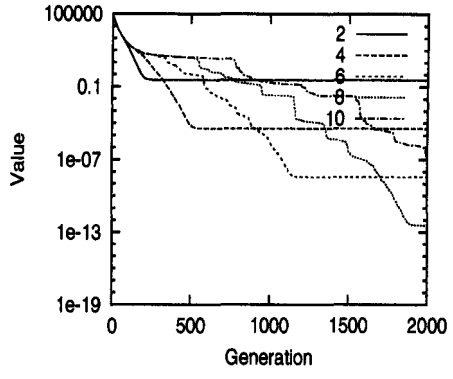
そして，これらの傾向は，$f_8$，$f_9$，$f_{10}$，$f_{11}$でも見られる．Fig. 5.8より$f_8$では下限が$10^{-4}$より小さい場合に100世代付近で，Fig. 5.9より$f_9$ではCESの下限が$10^{-4}$より小さい場合に500世代付近で，FESの下限が$10^{-4}$より小さい場合に200世代付近で，Fig. 5.10より$f_{10}$ではCESの下限が$10^{-4}$より小さい場合に200世代付近で，FESの下限が$10^{-4}$より小さい場合に100世代付近で，Fig. 5.11より$f_{11}$ではCESの下限が$10^{-4}$より小さい場合に200世代付近で，FESの下限が$10^{-4}$より小さい場合に100世代付近で，早期収束の傾向が顕著に見られ探索速度が落ちている．そのため，下限が小さくなるにしたがってRESの優位性が現れている．

## • 計算機実験結果 (まとめ)

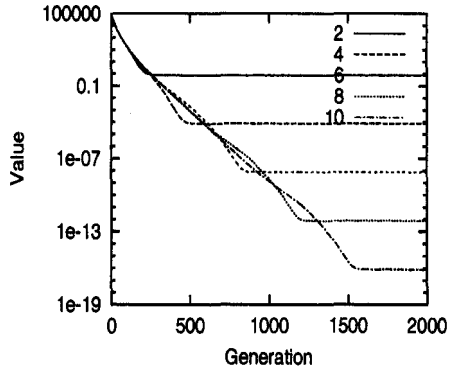
計算機実験結果によって得られた下限設定に対する頑健性をTable.5.2にまとめる．CESとFESは，全てのテスト関数において下限$\epsilon$によって異なる収束速度を示す．一方，RESは$\epsilon$の設定に依存せず，頑健であった．




(a) CES.

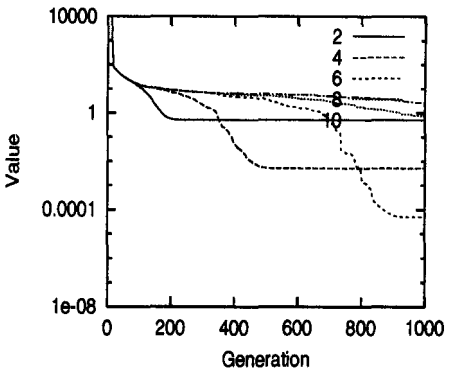



(b) FES.

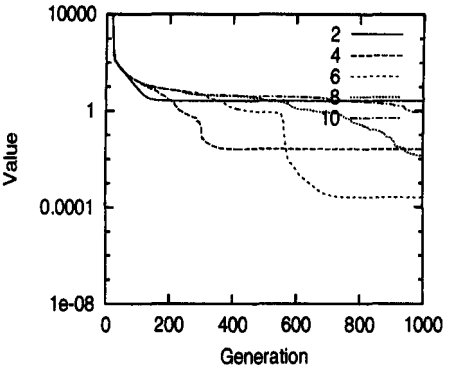



(c) RES.

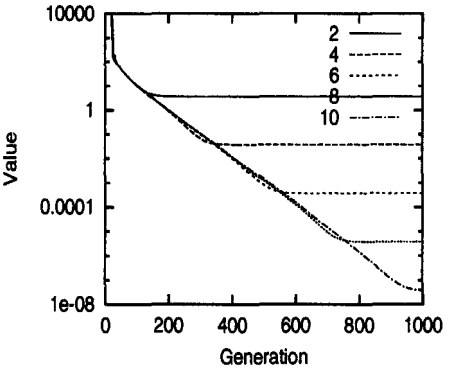
Fig. 5.1: Averaged best results on $f_1$ when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.




(a) CES.

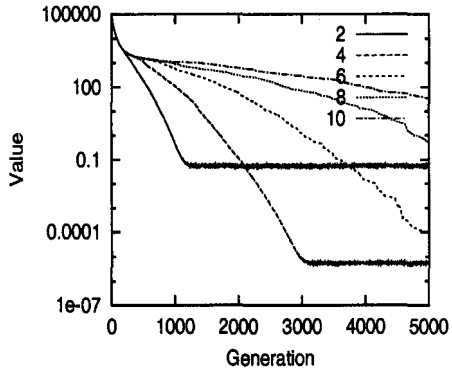



(b) FES.

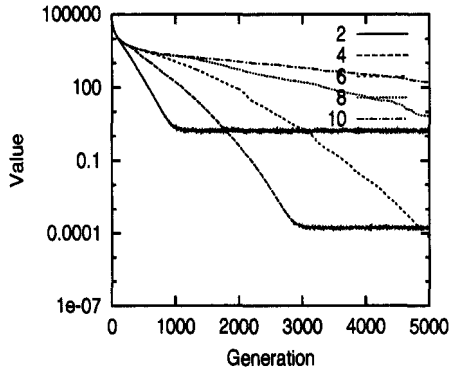



(c) RES.

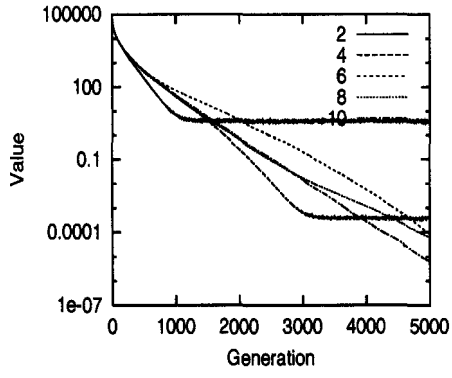
Fig. 5.2: Averaged best results on $f_2$ when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.




(a) CES.

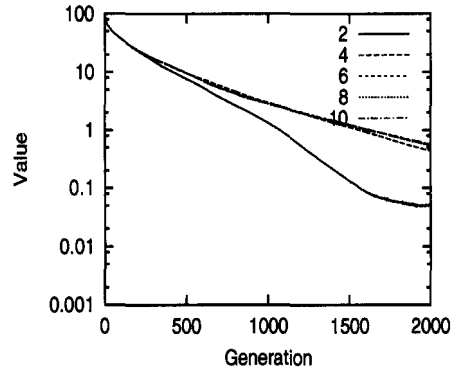



(b) FES.

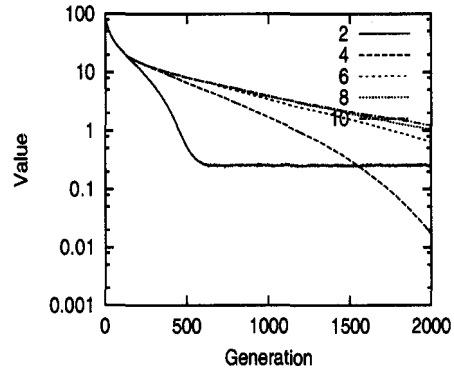



(c) RES.

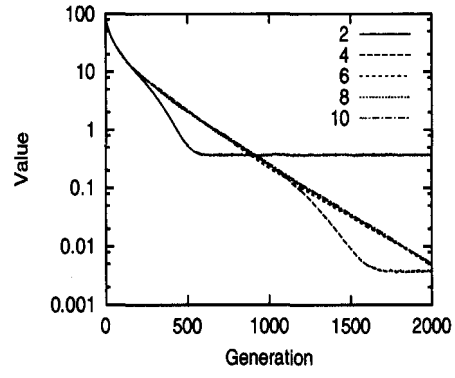
Fig. 5.3: Averaged best results on $f_3$ when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.




(a) CES.

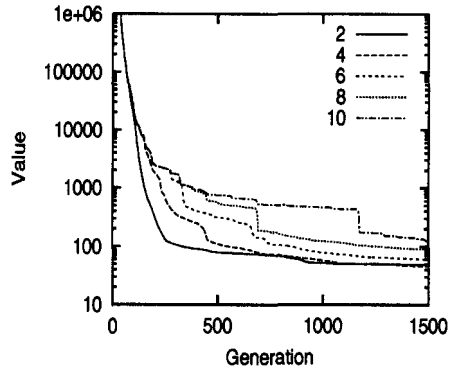



(b) FES.

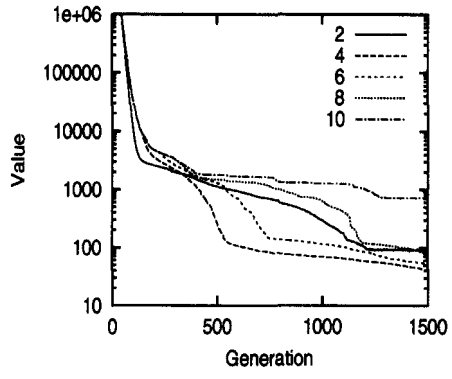



(c) RES.

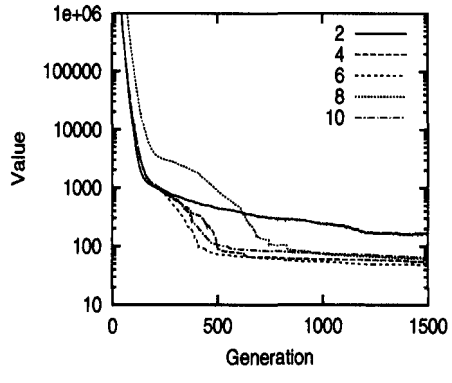
Fig. 5.4: Averaged best results on $f_4$ when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.




(a) CES.

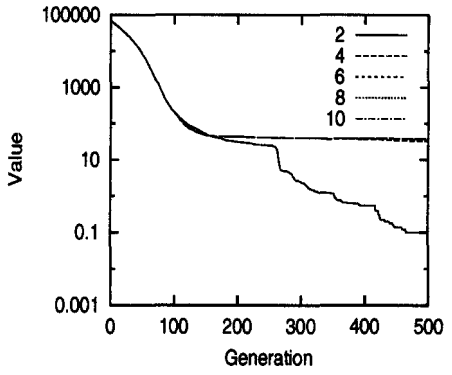



(b) FES.

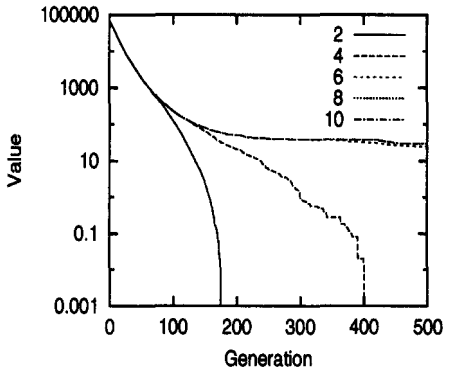



(c) RES.

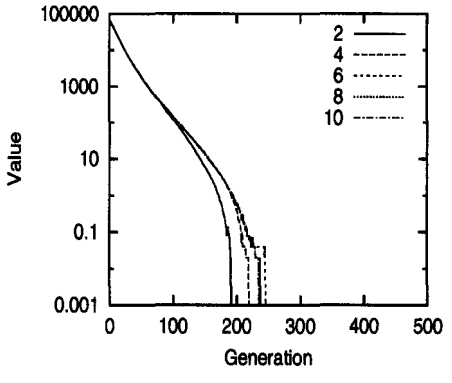
Fig. 5.5: Averaged best results on $f_5$ when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.




(a) CES.

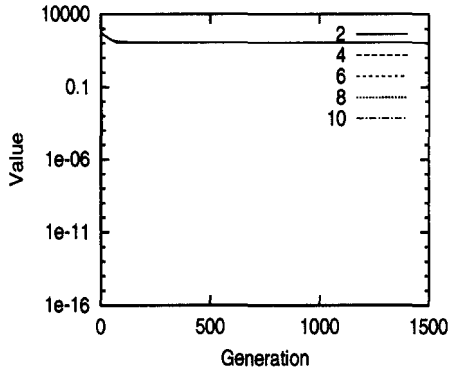



(b) FES.

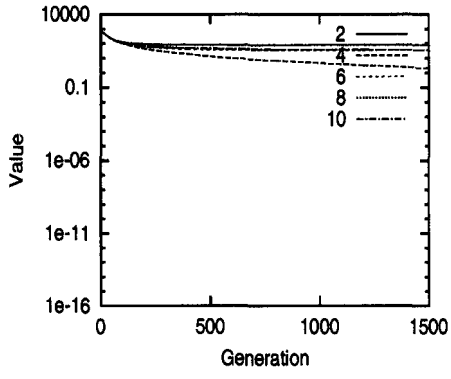



(c) RES.

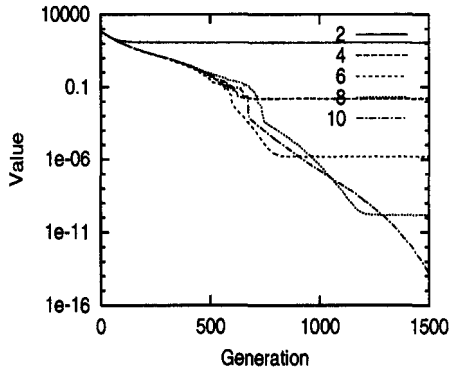
Fig. 5.6: Averaged best results on $f_6$ when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.




(a) CES.

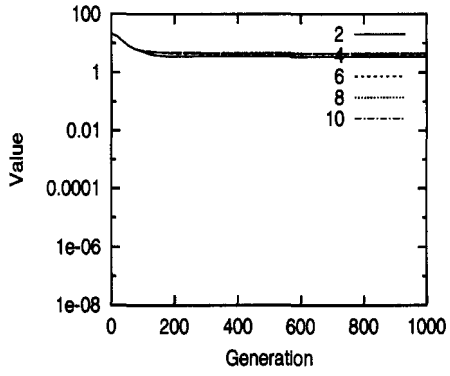



(b) FES.

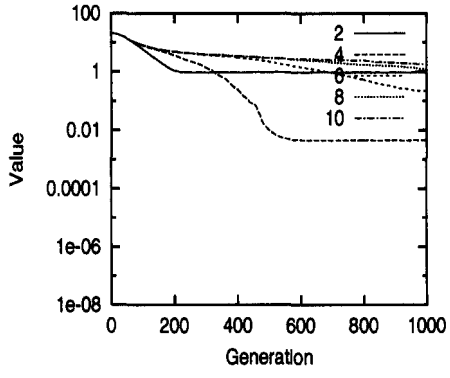



(c) RES.

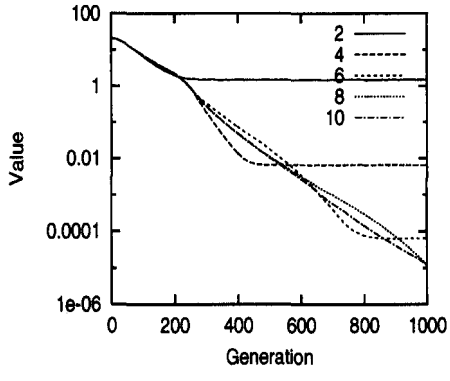
Fig. 5.7: Averaged best results on $f_7$ when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.




(a) CES.

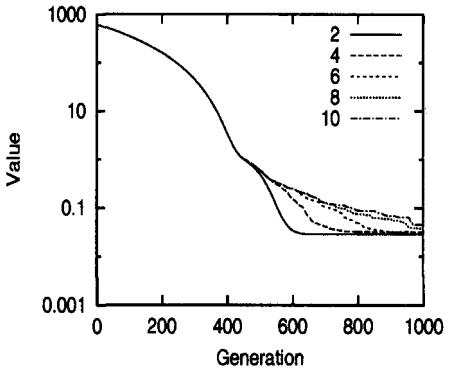



(b) FES.

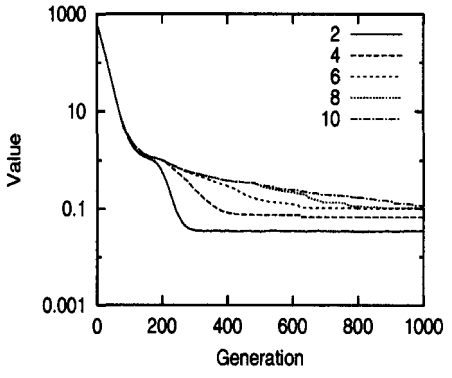



(c) RES.

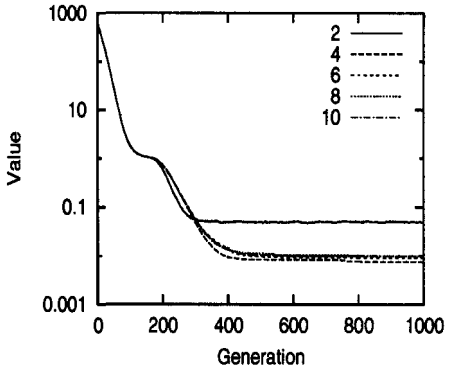
Fig. 5.8: Averaged best results on $f_8$ when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.




(a) CES.

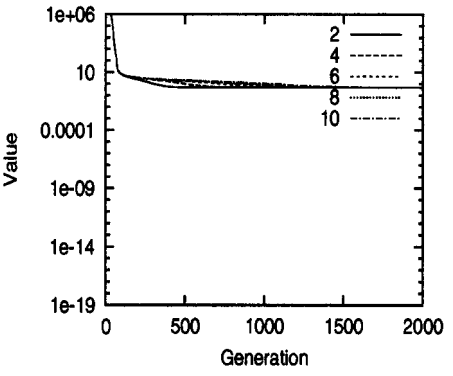



(b) FES.

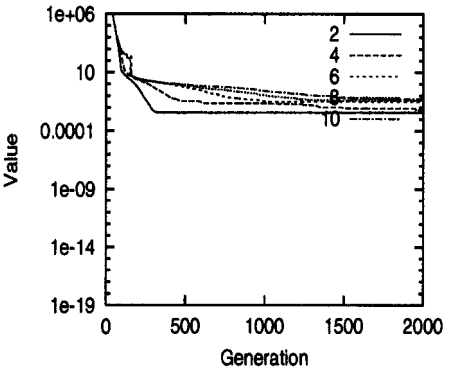



(c) RES.

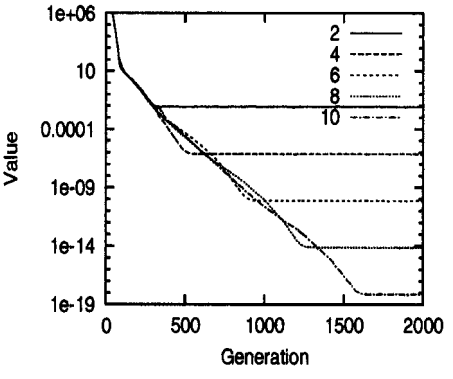
Fig. 5.9: Averaged best results on $f_9$ when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.




(a) CES.

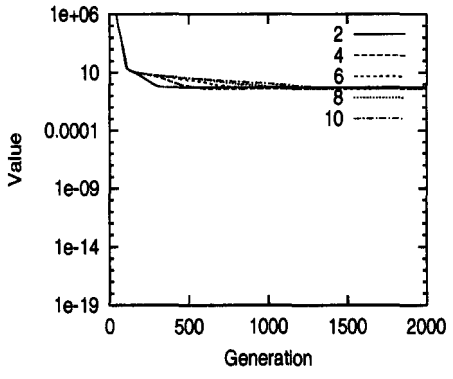



(b) FES.

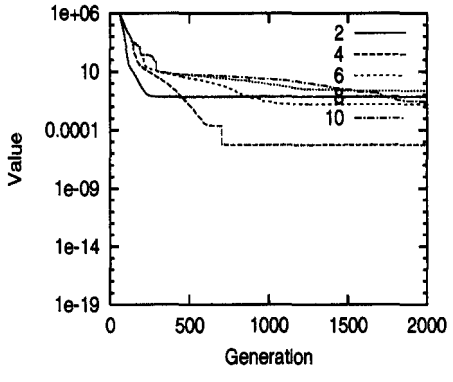



(c) RES.

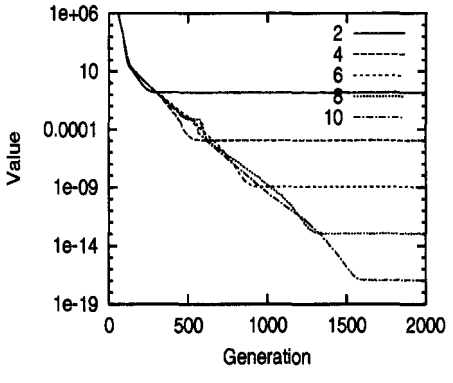
Fig. 5.10: Averaged best results on $f_{10}$ when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.




(a) CES.




(b) FES.




(c) RES.

Fig. 5.11: Averaged best results on $f_{11}$ when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.

### 5.2.3 進化ダイナミクス

#### • 次元スケールと進化ダイナミクス

問題の次元スケールによって，進化ダイナミクスがどのように影響を及ぼされるかを検証する．特に，以下の単峰性 ($f_1$) 及び多峰性 ($f_8$) の $n$ 次元実関数最適化ベンチマーク問題を取り扱う．これらの関数は大域最小値 0 を原点 (0,...,0) で持つ (Fig. 5.12).

(Hypersphere Function)

$$f_1(x) = \sum_{i=1}^{n} x_i^2 \qquad (5.1)$$
$$(-100 \leq x_i \leq 100)$$

(Ackley's Function)

$$f_8(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right)$$
$$-\exp(\tfrac{1}{n}(\textstyle\sum_{i=1}^{n}\cos 2\pi x_i)) + 20 + e \qquad (5.2)$$
$$(-32 \leq x_i \leq 32)$$

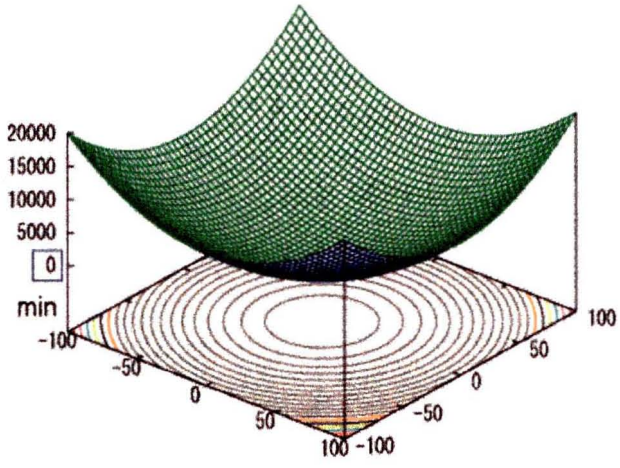
$(\mu, \lambda) = (30, 200)$ とし，戦略パラメータの下限，相関突然変異および組換えを用いないこととした．次元は $n$=1，5，10，15，20 に関して計算する．さらに，RES [89][90][77] では，$m = 5$ として，$g_{dup}$，$g_{del}$，$g_{inv}$ の適用確率 (P) を，それぞれ，$P(g_{dup}) = 0.6$，$P(g_{del}) = 0.3$，$P(g_{inv}) = 0.1$ とした．なお，CES と FES の実験は，RES で CES と FES に等価になるパラメータ値，すなわち，$P(g_{dup}) = 1$，$P(g_{del}) = 0$，$P(g_{inv}) = 0$ として行った．全ての計算機実験において初期集団は同一のものを用い，50 回試行を繰り返すこととした．$f_1$ は 2000 世代を，$f_8$ は 1000 世代を最終世代とした．

#### • 計算機実験結果 ($f_1$)

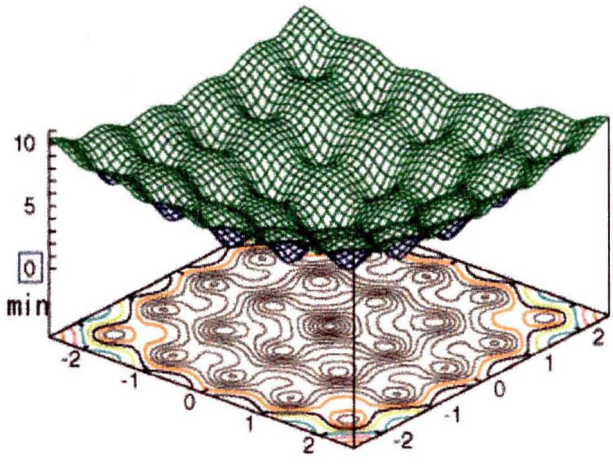
単峰性関数 $f_1$ の計算結果を Fig. 5.13 から Fig. 5.17 に示す．それぞれ，縦軸に関数値，横軸に世代をとった．

次元 $n$=1 と 5 では，CES と FES と RES に性能の差は見られない．次元 $n$=10 では，CES と FES は下限が $10^{-4}$ より小さい場合，100 世代付近で早期収束の傾向が見られる．一方，RES においては，下限によってそれほど探索速度の差異は無い．これらの傾向は次元 $n$=15 から 20 になるに従って顕著に見られる．次元 $n$=15 と 20 で，CES と FES は下限が $10^{-4}$ より小さい場合，100 世代付近で早期収束の傾向が見られるが，RES においては，下限によってそれほど探索速度の差異は無い．

#### • 計算機実験結果 ($f_8$)




(a) $f_1$ (Hypersphere Function).

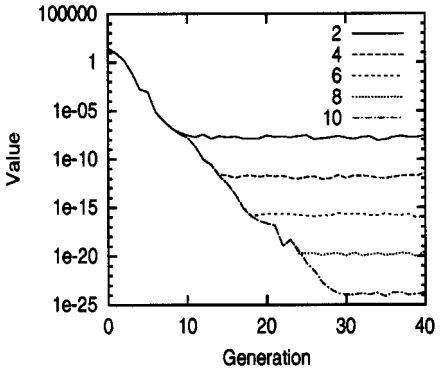



(c) $f_8$ (Ackley's Function).

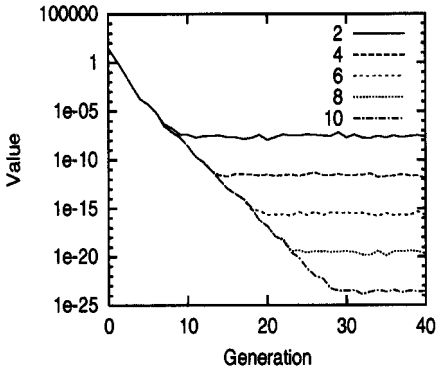
Fig. 5.12: The 2 dimensional landscape.

単峰性関数 $f_8$ の計算結果を Fig. 5.18 から Fig. 5.22 に示す．それぞれ，縦軸に関数値，横軸に世代をとった．

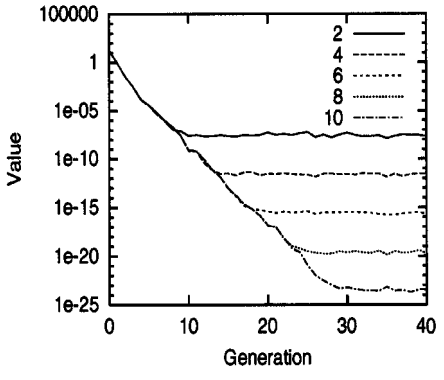
次元 $n$=1 では，CES と FES と RES に性能の差は見られない．次元 $n$=5 では，CES で，次元 $n$=10 では，FES で局所解から抜け出せない様子が分かる．一方，RES においては，局所解に陥ることなく大域最適解に近づくことができている．これらの傾向は次元 $n$=15 から 20 でも同様に見られる．




(a) CES.

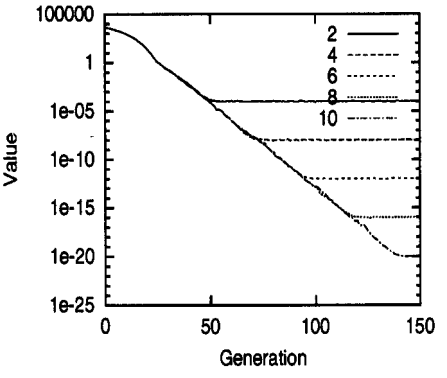



(b) FES.

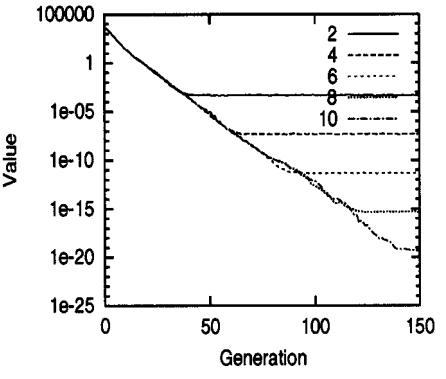



(c) RES.

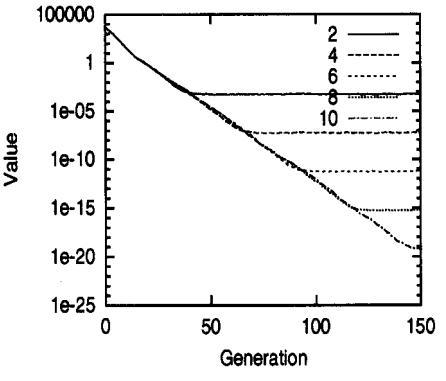
Fig. 5.13: Averaged best results on $f_1$ ($n$=1) when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.




(a) CES.

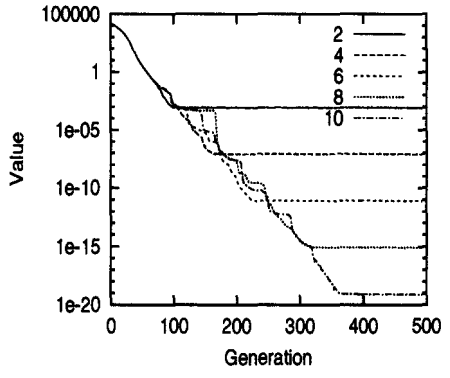



(b) FES.

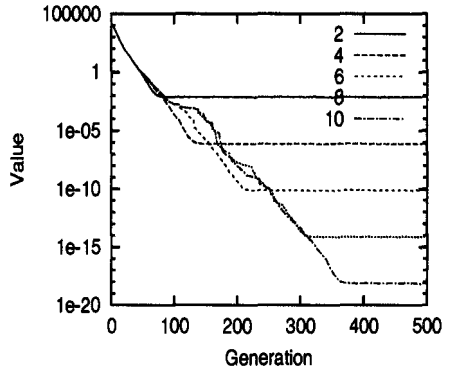



(c) RES.

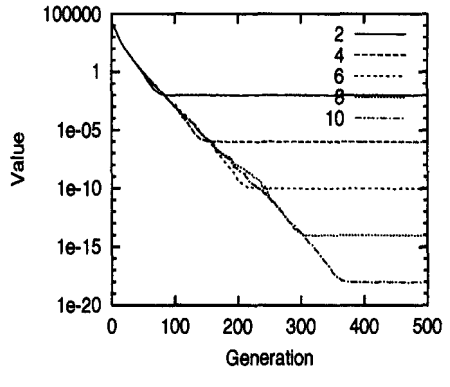
Fig. 5.14: Averaged best results on $f_1$ ($n$=5) when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.




(a) CES.

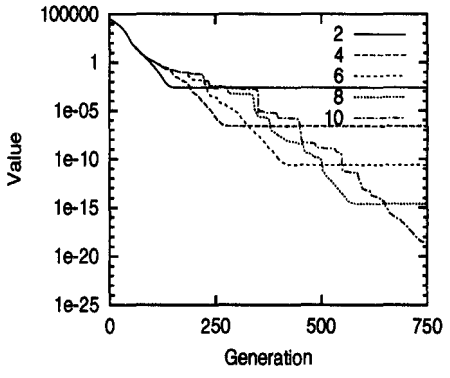



(b) FES.

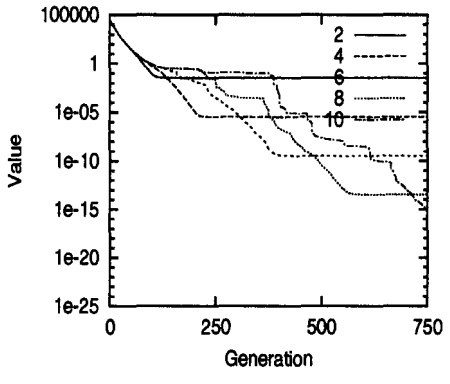



(c) RES.

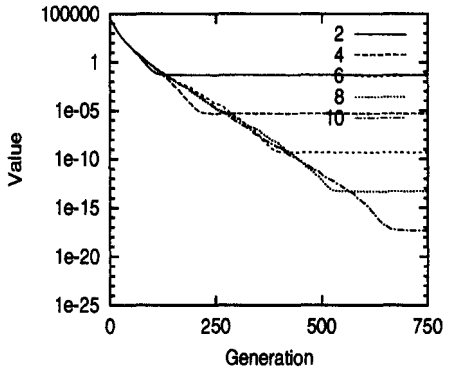
Fig. 5.15: Averaged best results on $f_1$ ($n$=10) when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.




(a) CES.

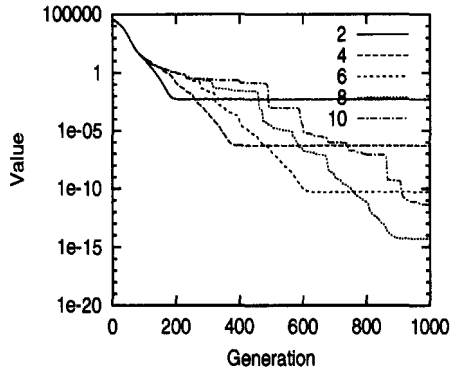



(b) FES.

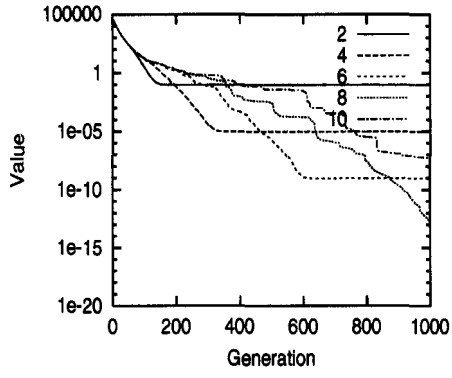



(c) RES.

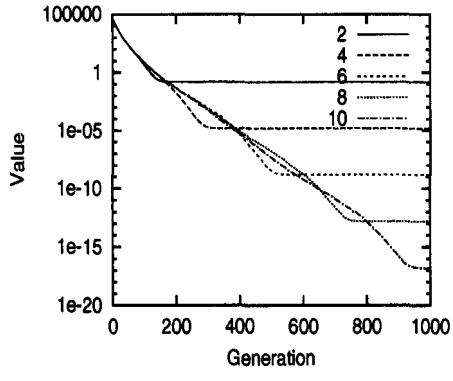
Fig. 5.16: Averaged best results on $f_1$ ($n$=15) when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.




(a) CES.

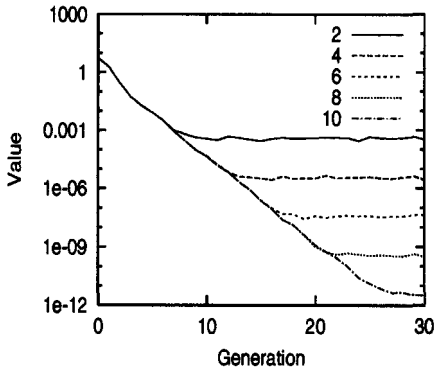



(b) FES.

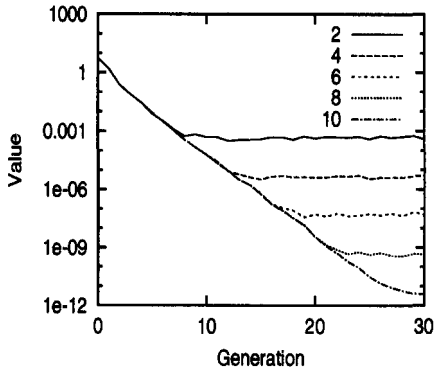



(c) RES.

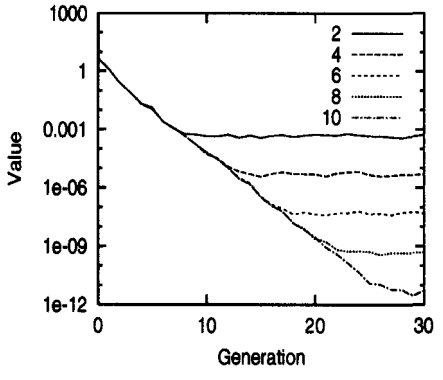
Fig. 5.17: Averaged best results on $f_1$ ($n$=20) when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.




(a) CES.

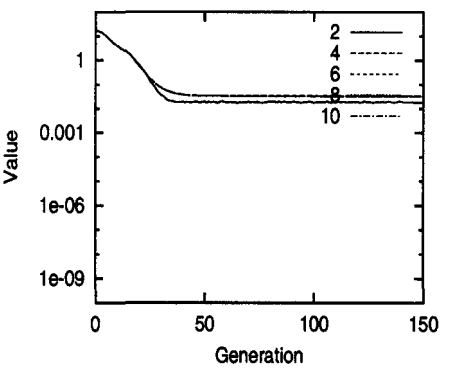



(b) FES.

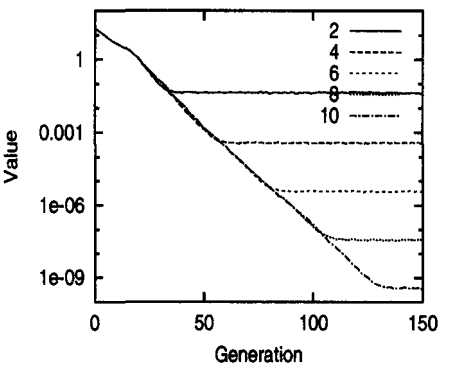



(c) RES.

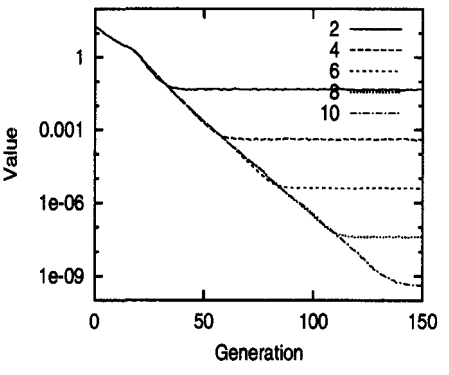
Fig. 5.18: Averaged best results on $f_8$ ($n$=1) when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.




(a) CES.

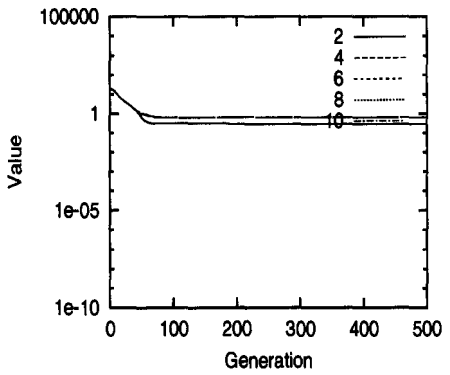



(b) FES.

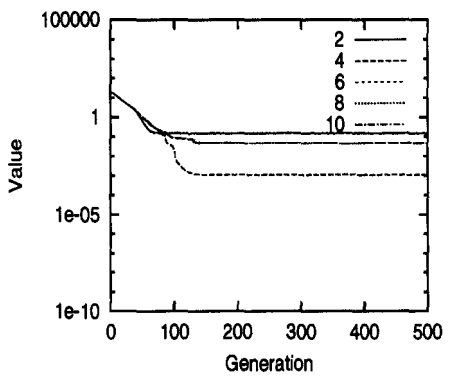



(c) RES.

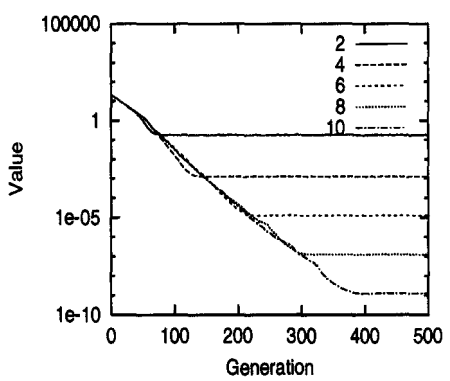
Fig. 5.19: Averaged best results on $f_8$ ($n$=5) when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.




(a) CES.

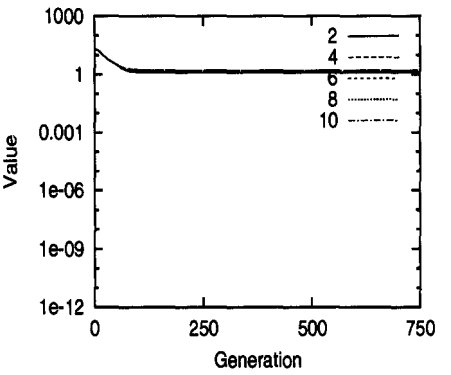



(b) FES.

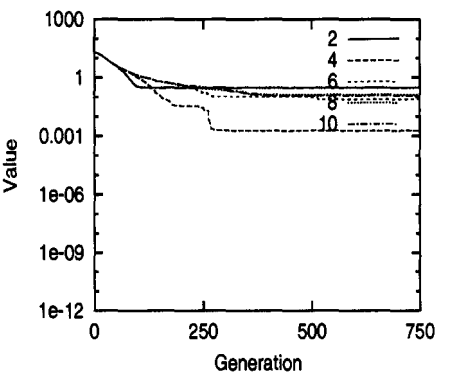



(c) RES.

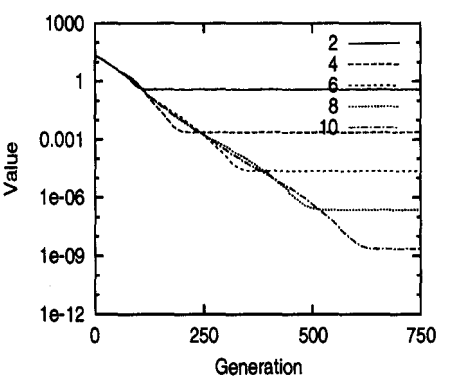
Fig. 5.20: Averaged best results on $f_8$ ($n$=10) when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.




(a) CES.

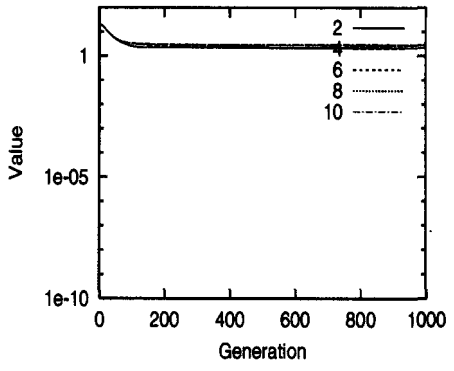



(b) FES.

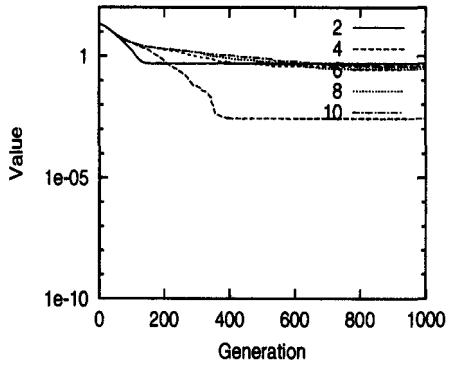



(c) RES.

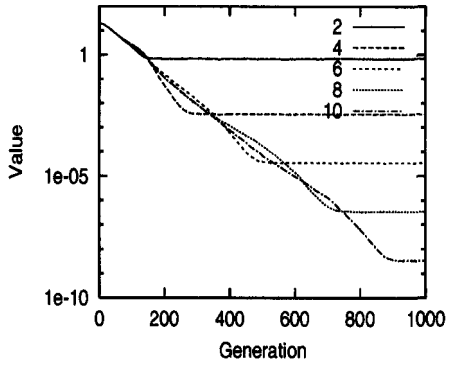
Fig. 5.21: Averaged best results on $f_8$ ($n$=15) when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.




(a) CES.




(b) FES.




(c) RES.

Fig. 5.22: Averaged best results on $f_8$ ($n$=20) when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.

## • 突然変異率と進化ダイナミクス

突然変異率の組合わせによって，RESの進化ダイナミクスがどのように影響を及ぼされるかを検証する．特に，以下の単峰性 ($f_1$) 及び多峰性 ($f_8$) の $n$ 次元実関数最適化ベンチマーク問題 ($n = 30$) を取り扱う．これらの関数は大域最小値0を原点 (0,...,0) で持つ．

(Hypersphere Function)

$$f_1(x) = \sum_{i=1}^{n} x_i^2 \qquad (5.3)$$
$$(-100 \leq x_i \leq 100)$$

(Ackley's Function)

$$f_8(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right)$$
$$-\exp(\tfrac{1}{n}(\textstyle\sum_{i=1}^{n}\cos 2\pi x_i)) + 20 + e \qquad (5.4)$$
$$(-32 \leq x_i \leq 32)$$

$(\mu, \lambda) = (30, 200)$ とし，戦略パラメータの下限，相関突然変異および組換えを用いないRESとした．戦略パラメータの上限 $\eta_{max}$ は3.0とし，$m = 5$ とした．提案手法における突然変異率の組合せ $(P(g_{dup}),\ P(g_{del}), P(g_{inv}))$ として以下のパラメータセットで計算する．

Case1 : $(P(g_{dup}),\ P(g_{del}), P(g_{inv})) = (0.2,0.1,0.05)$
Case2 : $(P(g_{dup}),\ P(g_{del}), P(g_{inv})) = (0.3,0.15,0.05)$
Case3 : $(P(g_{dup}),\ P(g_{del}), P(g_{inv})) = (0.4,0.2,0.1)$
Case4 : $(P(g_{dup}),\ P(g_{del}), P(g_{inv})) = (0.5,0.2,0.05)$
Case5 : $(P(g_{dup}),\ P(g_{del}), P(g_{inv})) = (0.6,0.3,0.1)$
Case6 : $(P(g_{dup}),\ P(g_{del}), P(g_{inv})) = (0.7,0.4,0.2)$
Case7 : $(P(g_{dup}),\ P(g_{del}), P(g_{inv})) = (0.8,0.4,0.15)$
Case8 : $(P(g_{dup}),\ P(g_{del}), P(g_{inv})) = (0.9,0.45,0.15)$
Case9 : $(P(g_{dup}),\ P(g_{del}), P(g_{inv})) = (1.0,0.0,0.0)$
Case10 : $(P(g_{dup}),\ P(g_{del}), P(g_{inv})) = (0.0,1.0,0.0)$
Case11 : $(P(g_{dup}),\ P(g_{del}), P(g_{inv})) = (0.0,0.0,1.0)$
Case12 : $(P(g_{dup}),\ P(g_{del}), P(g_{inv})) = (1.0,1.0,1.0)$

なお，Case1からCase8まではおよそ $(P(g_{dup}),\ P(g_{del}), P(g_{inv}))$=6:3:1 の比となるようなパラメータセットであり，遺伝的浮動の効果が得られる．Case9は遺伝的浮動効果

がほとんど得られないFESと等価になるパラメータ値である．Case9からCase11は1種類の操作だけであり，Case12の実験は，全ての操作を100%行う．全ての計算機実験において初期集団は同一のものを用い，50回試行を繰り返すこととした．$f_1$は2000世代を，$f_8$は1000世代を最終世代とした．

### • 計算機実験結果 ($f_1$)

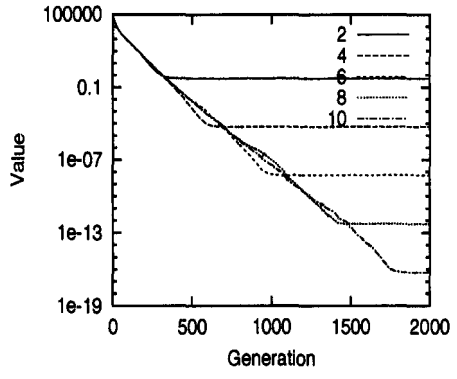
単峰性関数$f_1$の計算結果をFig. 5.23からFig. 5.24に示す．それぞれ，縦軸に関数値，横軸に世代をとった．

Case1からCase8では，下限の値によって収束速度は落ちることなく，小さい$\epsilon$を用いるほど精度良く大域最適解に近づくことができている．Case9では，下限が$10^{-4}$より小さい場合，200世代付近で早期収束の傾向が顕著に見られ収束速度が落ちている．Case10とCase12では，初期の段階で探索が収束している．Case11では，下限が小さくなるに従い探索速度が落ちている．

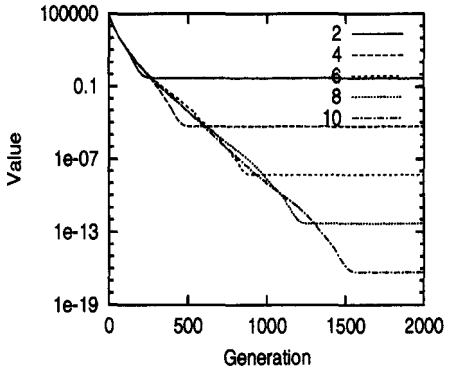
### • 計算機実験結果 ($f_8$)

多峰性関数$f_8$の計算結果をFig. 5.25からFig. 5.26に示す．それぞれ，縦軸に関数値，横軸に世代をとった．

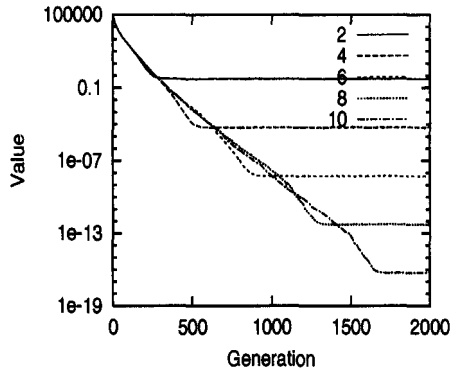
Case1からCase8では，下限の値によって収束速度は落ちることなく，小さい$\epsilon$を用いるほど精度良く大域最適解に近づくことができている．Case9では，下限が$10^{-4}$より小さい場合，100世代付近で早期収束の傾向が顕著に見られ収束速度が落ちている．Case10とCase12では，初期の段階で探索が収束している．Case11では，Case1からCase8に比較して探索速度が落ちている．



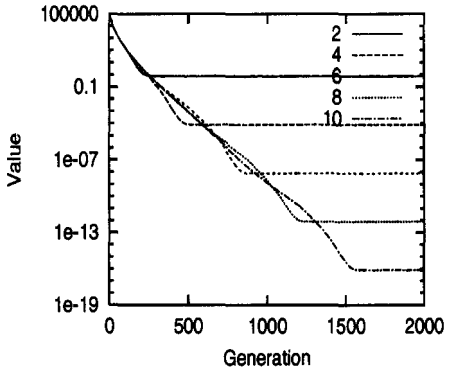

(a) RES (Case1).



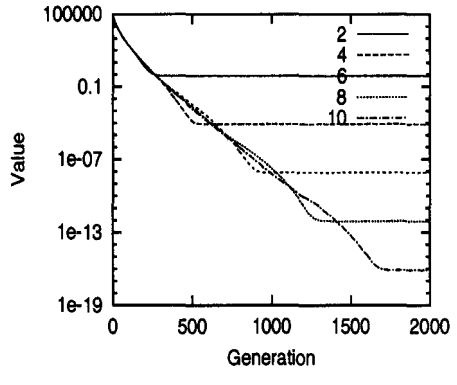

(a) RES (Case4).



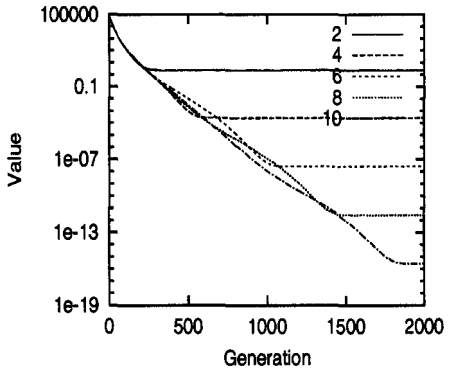

(b) RES (Case2).




(b) RES (Case5).

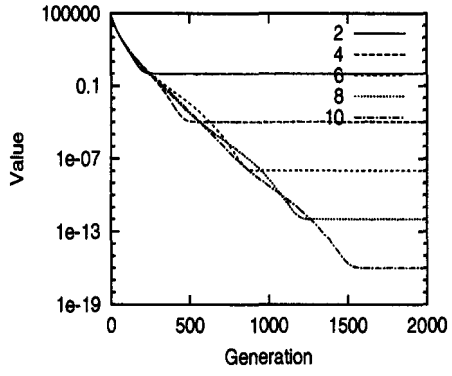



(c) RES (Case3).

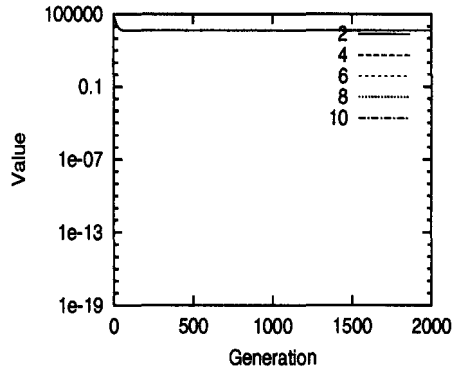



(c) RES (Case6).

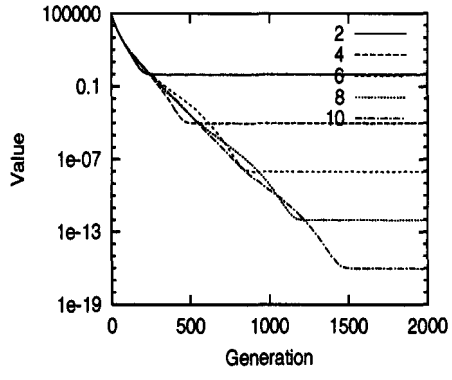
Fig. 5.23: Averaged best results on $f_1$ when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.



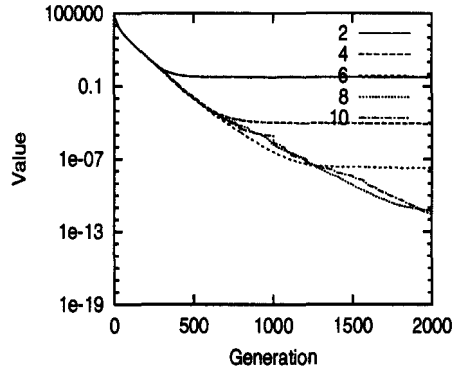

(a) RES (Case7).



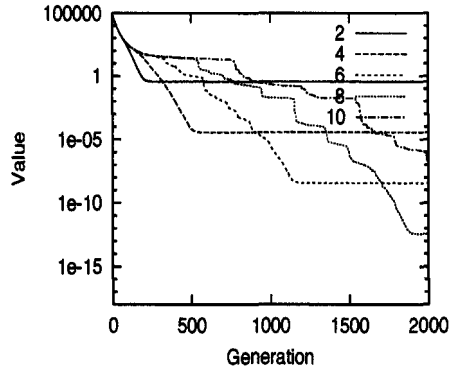

(a) RES (Case10).



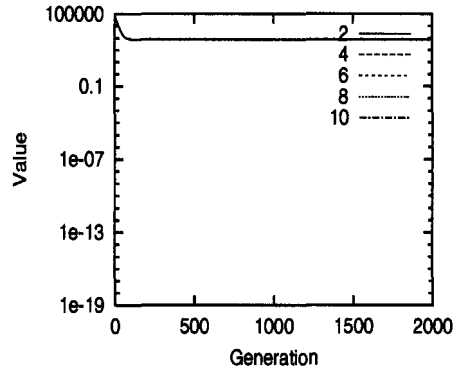

(b) RES (Case8).




(b) RES (Case11).

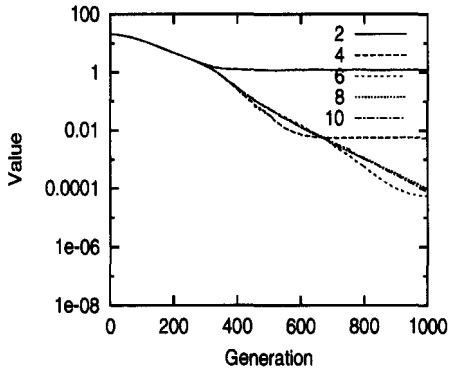



(c) RES (Case9).

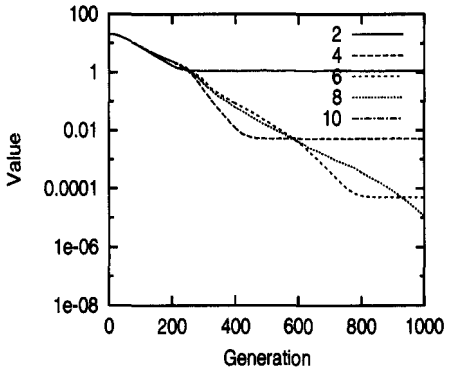



(c) RES (Case12).

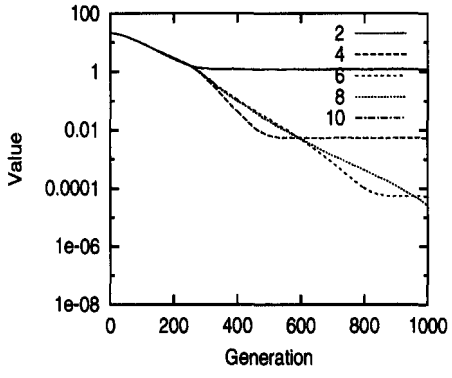
Fig. 5.24: Averaged best results on $f_1$ when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.



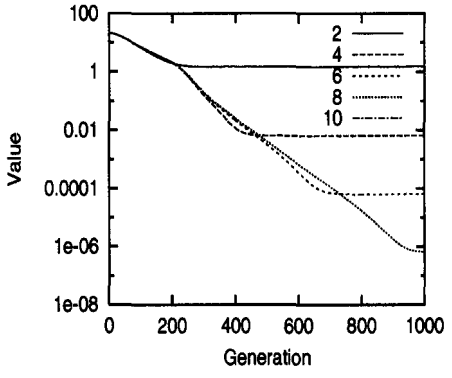

(a) RES (Case1).



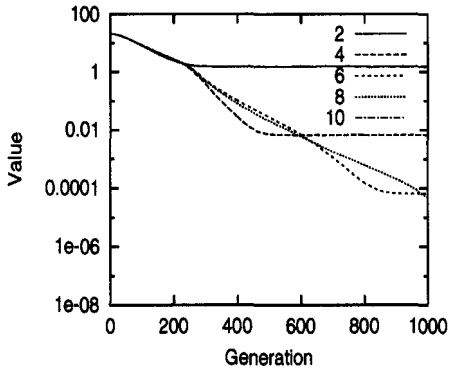

(a) RES (Case4).



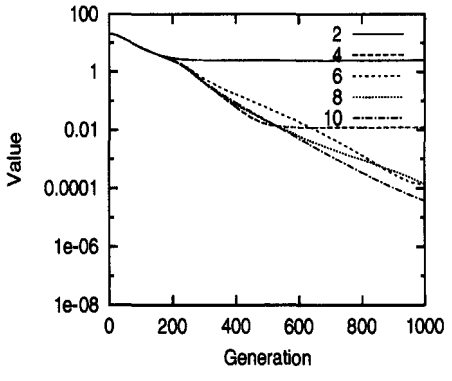

(b) RES (Case2).




(b) RES (Case5).

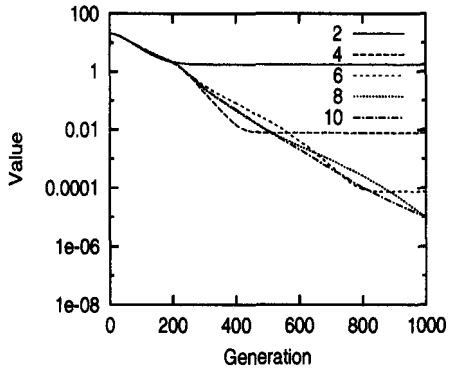



(c) RES (Case3).

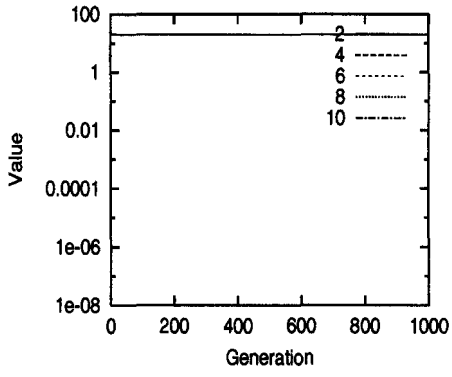



(c) RES (Case6).

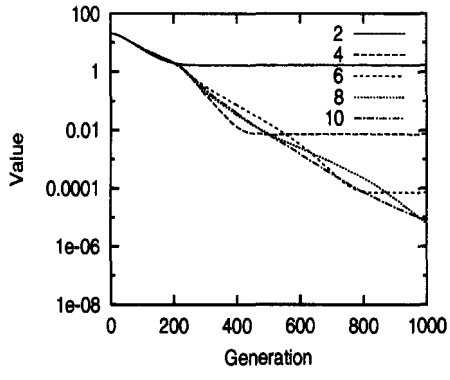
Fig. 5.25: Averaged best results on $f_8$ when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.



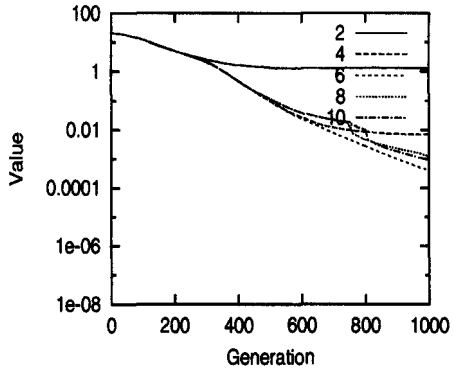

(a) RES (Case7).



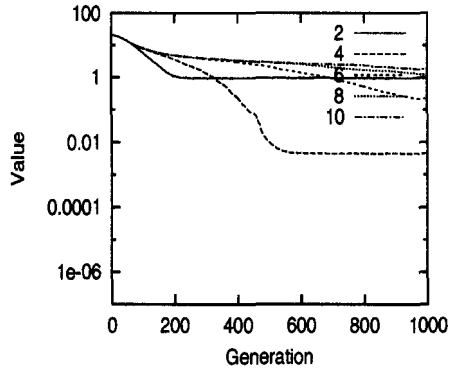

(a) RES (Case10).



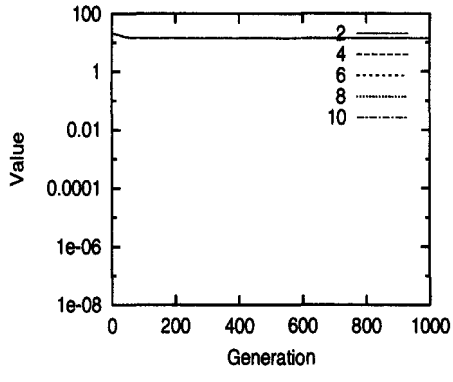

(b) RES (Case8).




(b) RES (Case11).




(c) RES (Case9).




(c) RES (Case12).

Fig. 5.26: Averaged best results on $f_8$ when the lower bounds are $10^{-2}$, $10^{-4}$, $10^{-6}$, $10^{-8}$ and $10^{-10}$.

### 5.2.4 探索特徴 (まとめ)

以上より，CES，FES，RESの探索特徴を以下のようにまとめることができる．

- 単峰性，多峰性関数に関係なくCES/FESの最適化能力は，下限の大きさに敏感であるが，RESは，下限にほとんど無関係であり，高速に最終的な解に落ち着く．
- 単峰性，多峰性関数に関係なく停滞状態になるまでに必要な世代数は，下限がある程度大きい時にはほとんど変わりないが，小さくなるに従いFESとRESの差が極端に大きくなり，最終的に必要な世代数はCES > FES > RESとなる．
- 単峰性関数の場合，小さい下限 を使用しても，非常に大きな世代数 (例えば，$f_1$では，下限が$10^{-10}$なら5000世代以上) さえ用いれば，RESで得られる精度がCESやFESでも得られることもある．
- 多峰性関数の場合，解の精度も探索速度もRES > FES > CES となる．

問題の次元スケールと進化ダイナミクスの関係を以下のようにまとめることができる．

- 単峰性，多峰性関数に関係なくある次元をこえると，遺伝的浮動の効果が得られる場合と得られない場合で，進化ダイナミクスに違いが見られる．

RESの突然変異率と進化ダイナミクスの関係を以下のようにまとめることができる．

- 単峰性，多峰性関数に関係なく遺伝的浮動の効果が得られる場合，突然変異率 ($P(g_{dup})$，$P(g_{del})$, $P(g_{inv})$) のパラメータセット (0.2,0.1,0.05) から (0.9,0.45,0.15) に対し非常に頑健であることが分かった．
- 単峰性，多峰性関数に関係なく遺伝的浮動の効果がほとんど得られない場合，収束速度が落ちる．
- 単峰性，多峰性関数に関係なく3種類の突然変異操作が適度に相互作用する場合，遺伝的浮動の効果がうまく発揮されやすい．

Table. 5.3: Summary of experiments for noisy test functions.

| | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ | $f_8$ | $f_9$ | $f_{10}$ | $f_{11}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| CES | × | × | × | × | × | × | × | × | × | × | × |
| FES | × | × | × | × | × | × | × | × | × | × | × |
| RES | × | × | ○ | × | × | ○ | × | × | × | × | ○ |

## 5.3 ノイズを含むテスト関数を用いた計算機実験

目的変数以外に含まれるノイズを想定したテスト関数に関する性能比較実験を行う. [12][58] にならい，平均0，標準偏差 $\sigma_\delta$ の標準正規分布に従うノイズ $N(0,\sigma_\delta)$ を加え，以下のようにテスト関数を定義する：

$$F(\boldsymbol{x}) = f(\boldsymbol{x}) + N(0,\sigma_\delta) \tag{5.5}$$

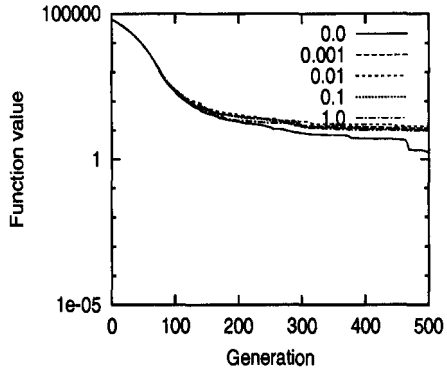
ここで，$\sigma_\delta$ をノイズレベルと呼び，ノイズレベルを $\sigma_\delta$={0.0，0.001，0.01，0.1，1.0} の値に設定し，$f(\boldsymbol{x})$ としてTable.4.2に示す11のテスト関数を用いて計算機実験を行った．$f_1$ から $f_6$ までは単峰性，$f_7$ から $f_{11}$ は多峰性関数である．すべてのテスト関数は30次元の探索空間を持ち，ノイズが存在しない場合の大域最小値は0である．

CESとFESおよびRESはともに，[117][119] に従って，$(\mu,\lambda) = (30,200)$，相関突然変異および組換えを用いないこととした．戦略パラメータの上限 $\eta_{max}$ は探索範囲が比較的狭い $f_7$ のみ 1.0，その他では3.0とし，下限 $\epsilon$ を設定しないで，それぞれ50回試行を繰り返した．さらに，RES [89][90][77] では，$m = 6$ として，$g_{dup}$，$g_{del}$，$g_{inv}$ の適用確率 (P) を，それぞれ，$P(g_{dup}) = 0.6$，$P(g_{del}) = 0.3$，$P(g_{inv}) = 0.1$ とした．なお，CESとFESの実験は，RESでCESとFESに等価になるパラメータ値，すなわち，$P(g_{dup}) = 1$，$P(g_{del}) = 0$，$P(g_{inv}) = 0$ として行った．

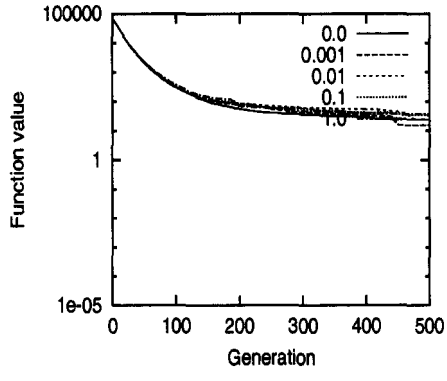
### • 計算機実験結果

テスト関数 $f_1$ から $f_{11}$ に関して，計算結果をFig. 5.27からFig. 5.37に示す．それぞれ，縦軸には50回の試行によって得られた親集団の平均関数値をとり，横軸には世代数をとったものである．

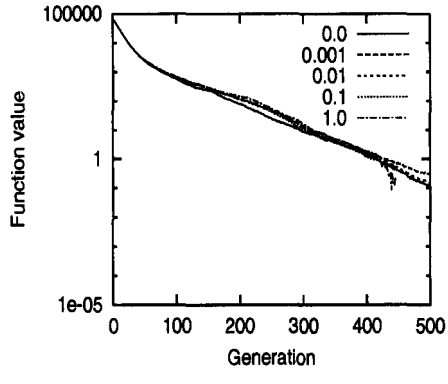
ノイズに対する頑健性をまとめたTable.5.3にも示すように，CESとFESは全てのテスト関数においてノイズの影響を受けた．一方，RESは $f_3$，$f_6$，$f_{11}$ のテスト関数でノイズに対して頑健であった．以上の結果より，RESはCES，FESに比べてノイズに対して頑健であると言える．しかしながら，第4章で示したEPの結果に比べると $(\mu,\lambda)$-ESはノイズの影響を受けやすいと言える．




(a) CES.

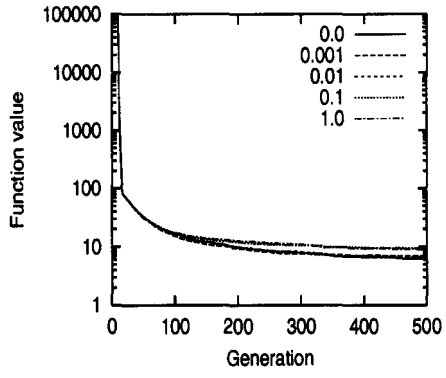



(b) FES.

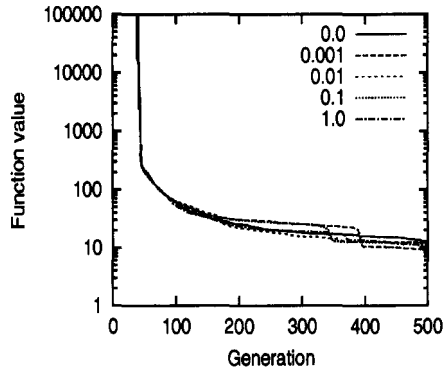



(c) RES.

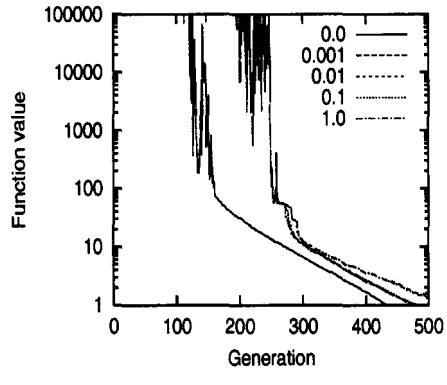
Fig. 5.27: Averaged best results on $f_1$ with input noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.




(a) CES.

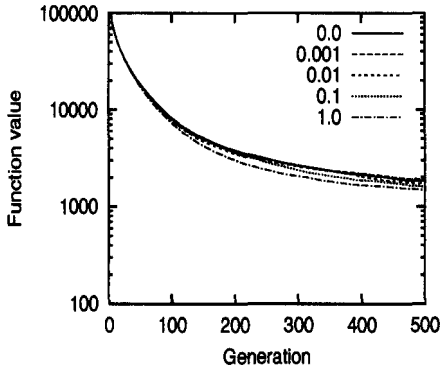



(b) FES.

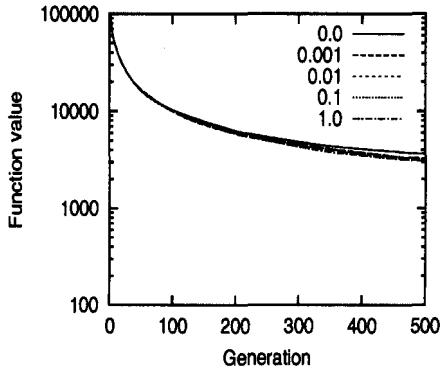



(c) RES.

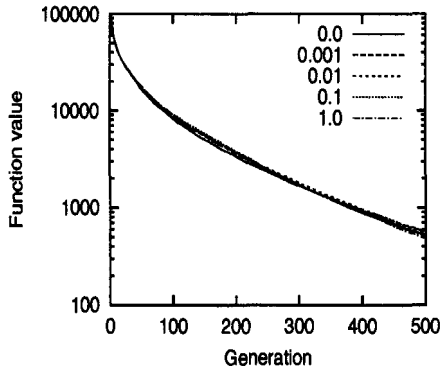
Fig. 5.28: Averaged best results on $f_2$ with input noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.




(a) CES.

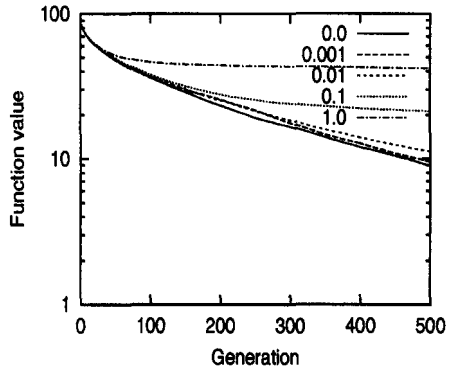



(b) FES.

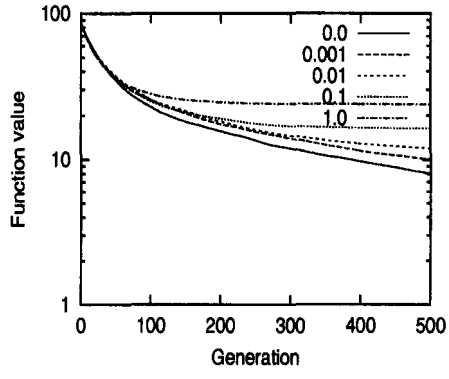



(c) RES.

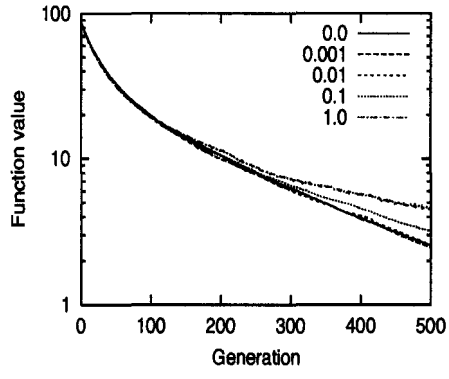
Fig. 5.29: Averaged best results on $f_3$ with input noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.




(a) CES.

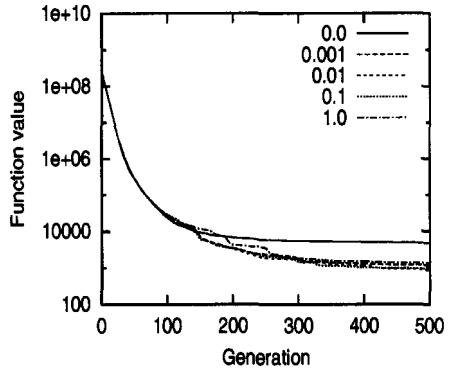



(b) FES.

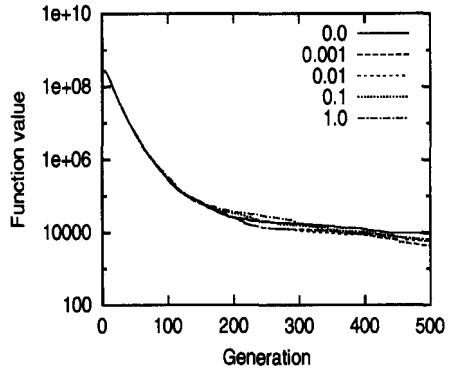



(c) RES.

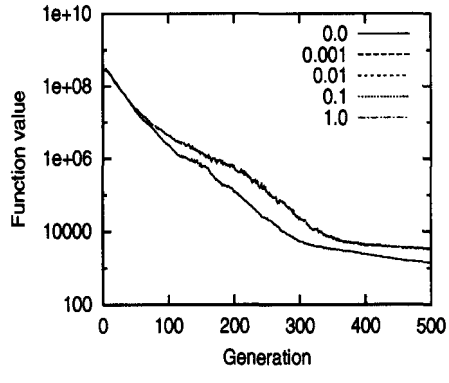
Fig. 5.30: Averaged best results on $f_4$ with input noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.




(a) CES.

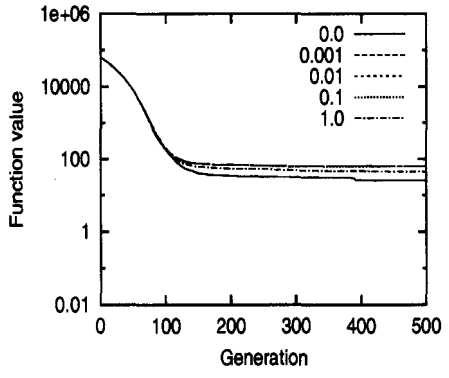



(b) FES.

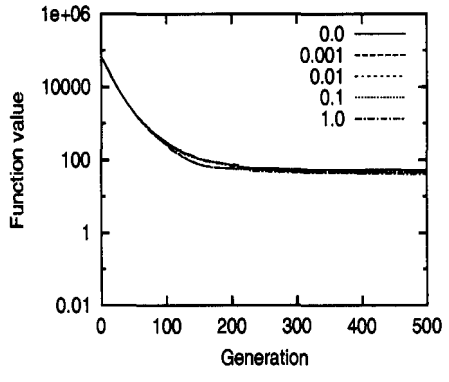



(c) RES.

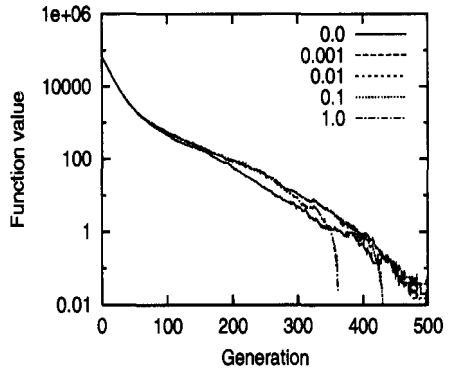
Fig. 5.31: Averaged best results on $f_5$ with input noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.




(a) CES.

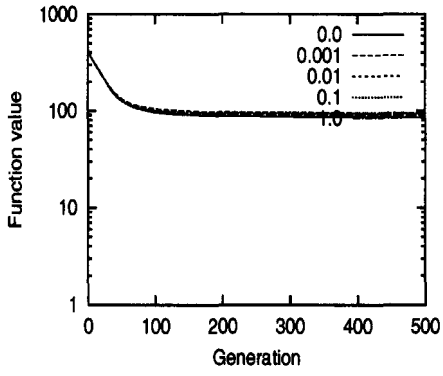



(b) FES.

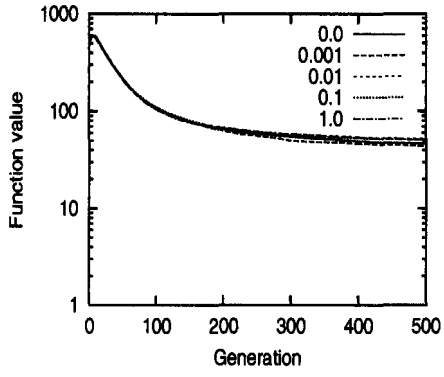



(c) RES.

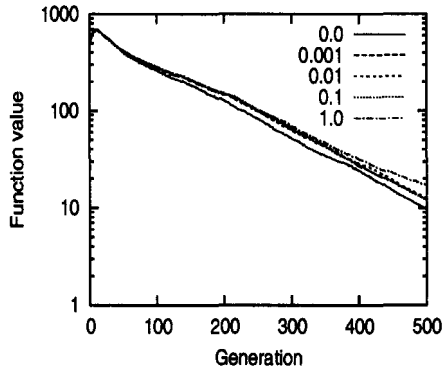
Fig. 5.32: Averaged best results on $f_6$ with input noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.




(a) CES.

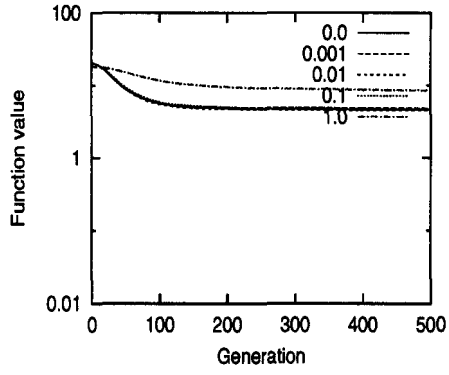



(b) FES.

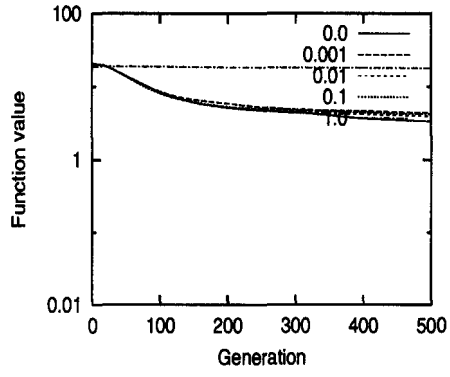



(c) RES.

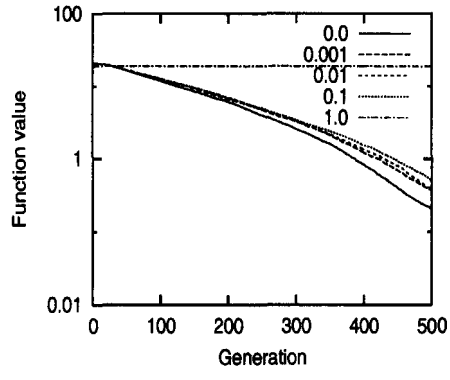
Fig. 5.33: Averaged best results on $f_7$ with input noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.




(a) CES.

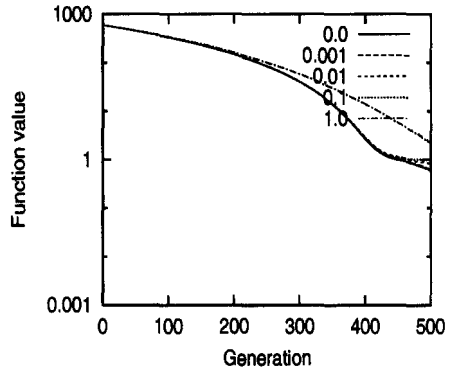



(b) FES.

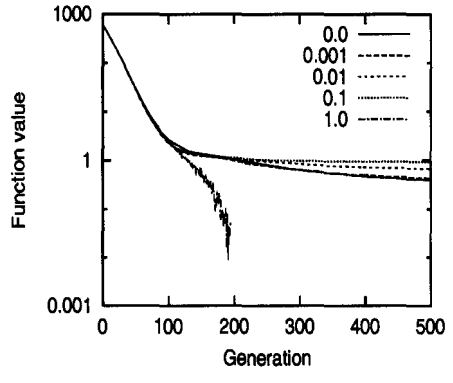



(c) RES.

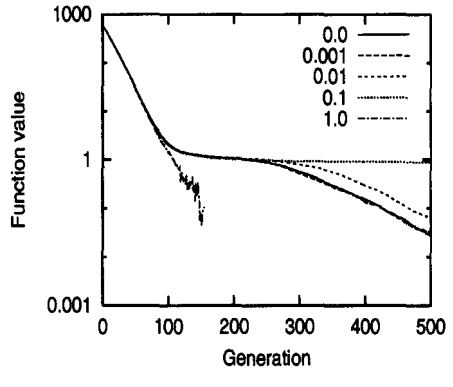
Fig. 5.34: Averaged best results on $f_8$ with input noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.




(a) CES.

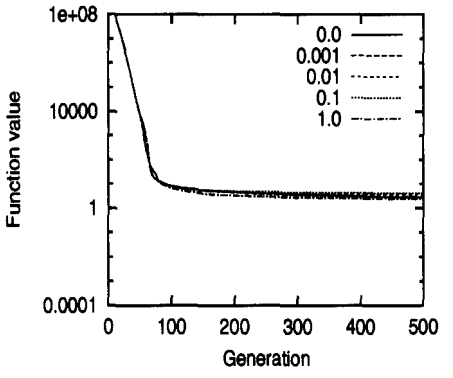



(b) FES.

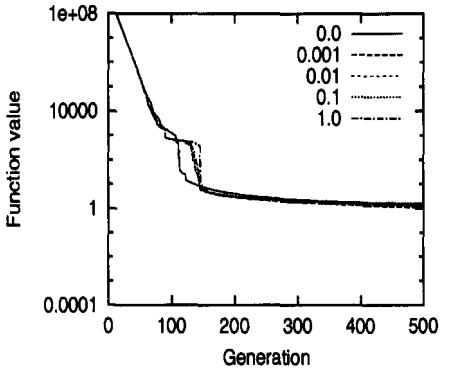



(c) RES.

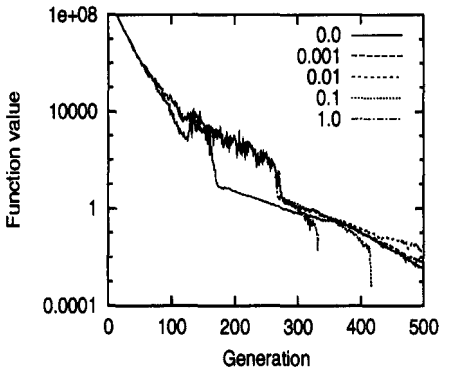
Fig. 5.35: Averaged best results on $f_9$ with input noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.




(a) CES.

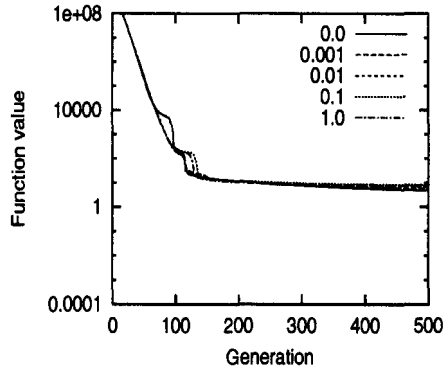



(b) FES.

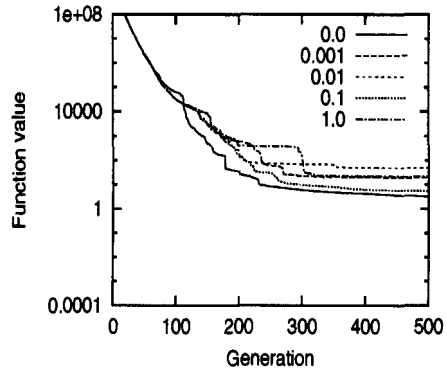



(c) RES.

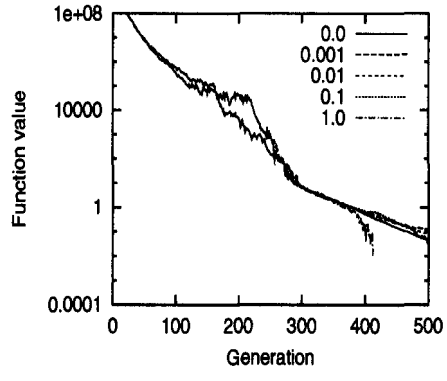
Fig. 5.36: Averaged best results on $f_{10}$ with input noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.




(a) CES.




(b) FES.




(c) RES.

Fig. 5.37: Averaged best results on $f_{11}$ with input noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.

## 5.4 Continuous-Time Recurrent Neural Networks (CTRNNs) への応用

工学的実問題への応用として，Evolutionary Robotics [61][50] における Evolutionary Artificial Neural Networks [115][122] を取り上げ，自律移動ロボットのナビゲーション問題を解く．2 次元空間内に障害物がある環境を，自律移動ロボットがスタート位置から移動し，光源に到達することをタスクとする．その際，ロボットには 1 ステップ毎に 10 個のセンサ値が入力され，左右 2 個の駆動輪のモータ値が出力されるものとする．入出力の写像関係は Continuous-Time Recurrent Neural Networks (CTRNNs) [18][19][20][106] で表現し，その設計を ES で行い，性能を検証する．

本節で対象とするシミュレータ環境を，Fig. 5.39 に示す．このシミュレータは Khepera ロボットを対象に，Sussex アプローチの下にミニマルシミュレーションに基づいて構築した [62][67][96]．ロボットは，1 ステップ毎に入力 $x_i(i = 0, 1, \ldots, 10 : 0 \leq x_i \leq 1.0)$ を感知し左右 2 個の駆動輪のモータ出力 $y_j(j = 0, 1 : 0 \leq y_j \leq 1.0)$ を決定する．入力は Fig. 5.40 の位置にある 8 個の測距センサから得られる入力 $x_i(i$=0,1,...,7) と前方 2 個の光センサから得られる入力 $x_i(i$=8,9) であり，Bias を入力 $x_i(i$=10) とした．ロボットのスタート位置は固定されているが，配置角度はランダムに設定され，ロボットの入力 $x_i'$ ならびに最終的な出力 $y_j'$ には，Gauss 分布から得られるノイズ $N(0, 1)$ $(0 \leq \delta \leq 1.0)$ が含まれるものとした:

$$x_i' = x_i + N(0, 1) \tag{5.6}$$

$$y_j' = 18.0 \times y_j - 10.0 + 2.0 \times N(0, 1) \tag{5.7}$$

ロボットのコントローラとして CTRNNs を用いこれを ($\mu$,$\lambda$)-ES を用いて進化的に設計する．CTRNNs は以下のように定式化される：

$$\tau_i \dot{\gamma}_i = -\gamma_i + \sum_{j=1}^{N} w_{ji} \sigma(g_j(\gamma_j + \theta_j)) + I_i, i = 1, \ldots, N \tag{5.8}$$

ただし，$\gamma$ はそれぞれのニューロンの状態，$\tau$ は時定数，$w_{ji}$ は $j$ 番目と $i$ 番目のニューロン間の重み，$g$ はゲイン，$\theta$ はバイアス項，$I$ は外部入力，$\sigma$ はシグモイド関数とする．なお，ニューロン数は5，$\gamma$ の初期値は0とし，オイラー法によって0.1刻みで積分した．スタートから $t$ ステップでゴールに到達した時に適応度を $1000 - t$ として評価する．なお，初期集団は $-5.0 \leq x_i(j) \leq 5.0$，$0 < \eta_i(j) \leq 3.0$ として CES と RES で同一のものを用い，100000 回の評価回数を目安に，($\mu$,$\lambda$)=(1,6) の場合は 15000 世代を，($\mu$,$\lambda$)=(3,20) の場合は 5000 世代を，($\mu$,$\lambda$)=(15,100) の場合は 1000 世代を，($\mu$,$\lambda$)=(30,200) の場合は 500 世代を最終世代とし，それぞれ 50 回繰り返して計算した．戦略パラメータの最大値を $\eta_{max}$=3.0 とした．RES のオペレータの適用確率は，$P(g_{dup}) = 0.6$，$P(g_{del}) = 0.3$，$P(g_{inv}) = 0.1$ とし，冗長な戦略パラメータ数を 5 に設定した．50 回の平均適応度を示

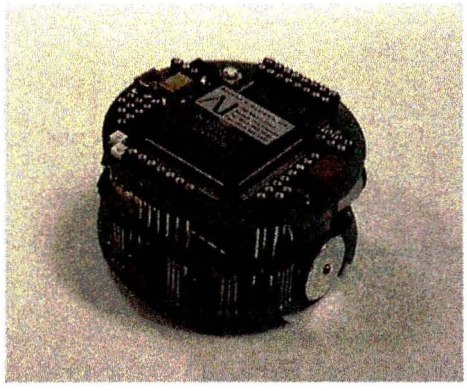


Fig. 5.38: Khepera Robot.

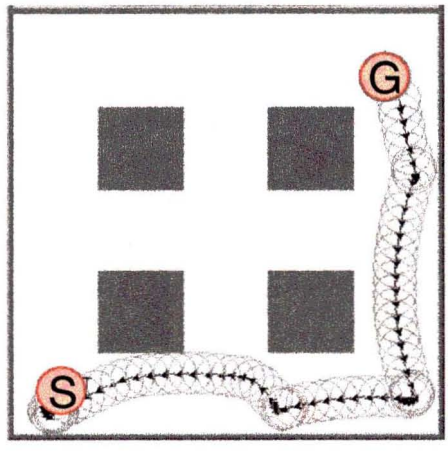



Fig. 5.39: Enviroment.

した Fig.5.41 から Fig.5.44 より, RES は CES に比べ個体数が増えるにつれて性能が向上している．CES では，(3,20) 以降，個体数を増やしても適応度はほぼ増加しないが，RES では (30,200) にすると平均適応度が 600 まで増加した．これは，適応度の高い領域が狭いため，個体数が (1,6) と極端に少ない場合はその領域に留まることが困難であり，(30,200) と個体数を増やすことによって，その領域に留まる個体と広く探索する個体とに分化する RES の特性が有効に働くためと考えられる．

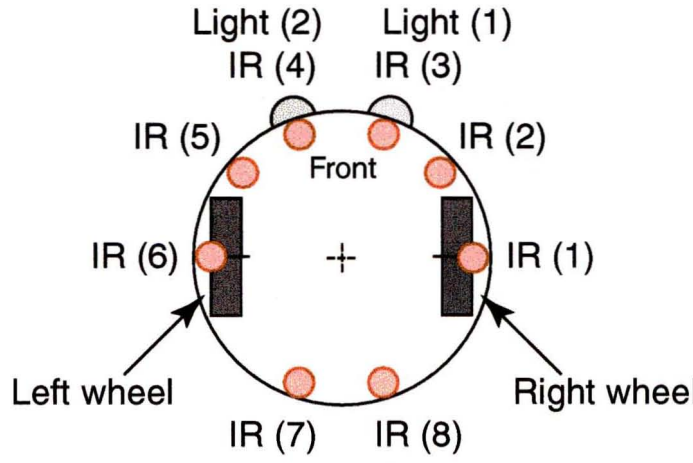



Fig. 5.40: Sensor position.

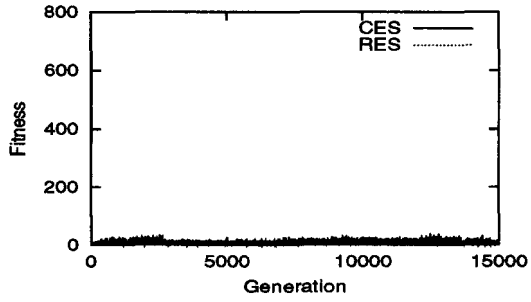



Fig. 5.41: The averaged fitness on (1,6)-ES for Continuous-Time Recurrent Neural Networks.

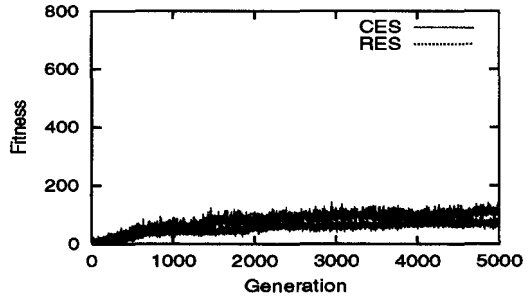



Fig. 5.42: The averaged fitness on (3,20)-ES for Continuous-Time Recurrent Neural Networks.

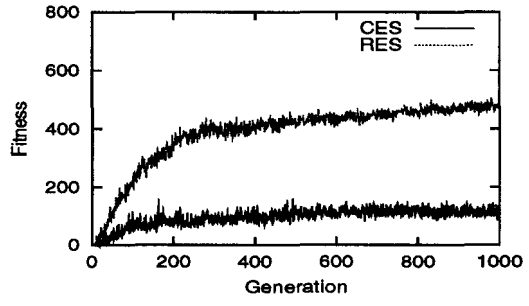



Fig. 5.43: The averaged fitness on (15,100)-ES for Continuous-Time Recurrent Neural Networks.

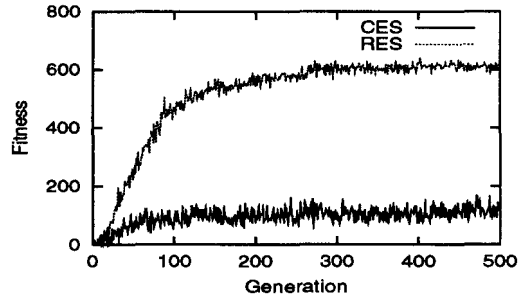



Fig. 5.44: The averaged fitness on (30,200)-ES for Continuous-Time Recurrent Neural Networks.

## 5.5 結言

本章では，$(\mu, \lambda)$-ESにおける自己適応の拡張法に関して，問題の次元に重点をおいて議論している．標準テスト関数を用いた計算機実験では，従来の進化戦略は，進化的プログラミングの場合と同様に，第2章で指摘した状態になりやすいことを明らかにした．これに対し，第3章で定式化した自己適応の拡張法を進化戦略に適用した手法を提案している．標準テスト関数を対象とした計算機実験により，提案手法は従来手法に比べて，戦略パラメータの下限値に対する頑健性が，向上することを実証した．提案手法の特徴を知るために突然変異率と問題の次元スケールによる進化ダイナミクスの変化を観測し，提案手法は問題の次元スケールと突然変異率のパラメータ値に対し頑健であることを確認した．また，ノイズを含むテスト関数に関してもノイズに対する提案手法の頑健性を実証した．さらに，工学的実問題への応用として，自律移動ロボットのナビゲーション問題における Continuous-Time Recurrent Neural Networks (CTRNNs) の進化的設計を取り扱った．この CTRNNs は，実数値の入出力を取り扱うことが可能であり，連続時間を対象にしている．そのため，回路規模に対して変数は指数関数的に増大し，標準テスト関数に比べて次元の高い最適化問題となる．この高次元の問題に対しても，従来手法に対する提案手法の性能の向上を計算機実験により実証した．以上より，第4章の進化的プログラミングと比べて確定的選択とエリート保存されない点が異なる進化戦略においても，自己適応の拡張法の妥当性を示したと言える．

# 第6章 $(\mu/\mu,\lambda)$-ESにおける自己適応の拡張とMulti-parent Recombinationによる補完

## 6.1 緒言

$(\mu/\rho,\lambda)$-ESには，Multi-parent Recombination (MPR) [45][47] としてIntermediate Recombination (IR) とDiscrete Recombination (DR) があるため，実数値変数と戦略パラメータのどちらか，もしくは両方に対して用いることができる．Schwefel [104][8] 等，性能は問題に依存するため，試行錯誤することを勧めながら，一般的にはDRを実数値変数に，IRを戦略パラメータに用いることを推奨している．しかしながら，Chang et al. [34] は$(\mu/\mu,\lambda)$-ESにおいて戦略パラメータにIRを用いるのみならず，戦略パラメータにDRを用いる手法の有効性を計算機実験により明らかにした．この研究では，11個の標準的なテスト関数によって，Classical ES (CES) の性能比較を行っている．

そこで，本章ではいくつかの$(\mu/\mu,\lambda)$-ESに関して，実数値変数と戦略パラメータの両方にIRを用いる場合，あるいはDRを用いる場合のMPRに関する性能を検証し，進化過程での集団の進化ダイナミクスを調べる．計算機実験では，ESの形式としてGauss型突然変異を用いるCES [104][13]，Cauchy型突然変異を用いるFast ES (FES) [117][119]，冗長個体表現を持つRobust ES (RES) [89][90][77] を採用する．なお本章では，Gauss型突然変異を用いるRESをgRES，Cauchy型突然変異を用いるRESをcRESとして参照する．そして，YaoとLiu [117][119] が用いた標準的なテスト関数のうち代表的な関数を取り上げ，それぞれのESにおけるMPRの性能を検証する．さらに，子孫の進化ダイナミクスがMPRによってどのように変化するかを主成分分析によって解析する．また，工学問題に応用する上で重要となるノイズに対する頑健性を検証した後に，実問題に応用しその有用性を検証する．

Table. 6.1: Contents of Section 6.

| 6章 | 6.2 | 6.3 | 6.4 |
|---|---|---|---|
| 問題 | 標準テスト関数 | ノイズを含むテスト関数 | ANNの設計問題 |
| 手法 | CES,FES,gRES,cRES | CES,FES,gRES,cRES | CES,FES,gRES,cRES |

## 6.2 標準テスト関数を用いた計算機実験

### 6.2.1 テスト関数と諸設定

MPR の有効性と進化過程の特徴を明らかにするために，Table.4.2 に示す 11 のテスト関数を用い，計算機実験を行った．これらは，それぞれ，Hypersphere 関数 ($f_1$)，Schwefel の問題 2.22($f_2$)，Schwefel の問題 1.2($f_3$)，Schwefel の問題 2.21($f_4$)，Rosenbrock 関数 ($f_5$)，Step 関数 ($f_6$)，Rastrigin 関数 ($f_7$)，Ackley 関数 ($f_8$)，Griewank 関数 ($f_9$)，Penalized 関数 (P8)($f_{10}$)，Penalized 関数 (P16)($f_{11}$) である．また，$f_1$ から $f_6$ までは単峰性，$f_7$ から $f_{11}$ は多峰性関数である．すべてのテスト関数は 30 次元の探索空間を持ち，大域最小値は 0 である．CES，FES および gRES，cRES は，Yao と Liu [117][119] に従って，$(\mu, \lambda) = (30, 200)$，相関突然変異を用いないこととした．戦略パラメータの上限 $\eta_{max}$ は探索範囲が比較的狭い $f_7$ のみ 1.0，その他では 3.0 とし，それぞれ 50 回試行を繰り返した．さらに，gRES と cRES では，$m = 5$ として，$g_{dup}$，$g_{del}$，$g_{inv}$ の適用確率 (P) を，それぞれ，$P(g_{dup}) = 0.6$，$P(g_{del}) = 0.3$，$P(g_{inv}) = 0.1$ とした [80][81].

なお，これらの推奨パラメータ値は，予備的な実験結果から決定した．また，同実験から，パラメータ値の変化に関して RES は非常に頑健であることが分かったため，パラメータ値チューニングにはそれほど注意を払っていない．それゆえ，詳細な実験を通してこれらを決定すれば，パフォーマンスは更に向上すると思われる．

### 6.2.2 基本探索性能

テスト関数 $f_1, \ldots, f_{11}$ の CES，FES，gRES，cRES による 50 試行の平均結果をそれぞれ Fig. 6.1 から Fig. 6.11 に示す．縦軸には最良個体の関数値の 50 回平均を，横軸には世代をとる．

The lower bound problem [90] のため CES は早い世代で探索がうまくいかないが，MPR を用いることによって $f_4$ の DD-CES，$f_7$ の II-CES 以外で性能が向上する．特に，II-CES は $f_1$，$f_3$，$f_5$，$f_6$，$f_8$，$f_9$，$f_{11}$ に対して有効であり，DD-CES は $f_7$，$f_{10}$ に対して有効であり，D-CES は $f_2$，$f_4$ に対して有効である．なお，CES は多峰性関数 $f_7$，$f_9$ で MPR を用いた場合も局所解に陥り，十分な探索能力を発揮していない．これらの結果は，従来研究 [34] 同様，実数値変数と戦略パラメータにどの Recombination を用いれば最も性能が向上するかは，問題の適応度景観に依存することを意味する．

FES は CES と異なり，II-FES は全ての関数で探索が進まないが，DD-FES，及び D-FES は性能が向上することが分かる．なお，DD-FES, D-FES は多峰性関数 $f_7$，$f_9$，$f_{10}$，$f_{11}$ で局所解に陥る．

gRES は FES と同様，II-gRES は全ての関数で探索が進まないことが分かる．一方，$f_3$ 以外の DD-gRES，及び全ての D-gRES で性能が向上する．また，全ての多峰性関

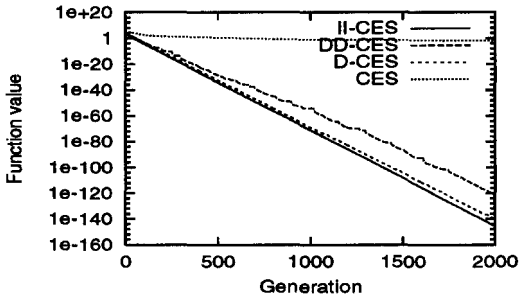
Table. 6.2: Summary of experiments for standard test functions.

| | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ | $f_8$ | $f_9$ | $f_{10}$ | $f_{11}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| CES | II | D | II | D | II | II | DD | II | II | DD | II |
| FES | D | D | *D* | D | D | D | DD | D | DD | DD | DD |
| gRES | D | D | D | D | D | D | D | D | D | D | D |
| cRES | D | D | D | D | D | D | D | D | DD | D | D |

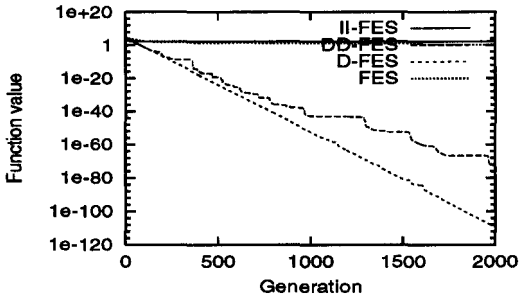
数においてDD-gRES，D-gRESとも大域最小値を発見する．

cRESはgRESと同様，II-cRES II-gRESは全ての関数で探索が進まないが，$f_3$，$f_5$を除くDD-cRES及び全てのD-cRESで性能が向上することが分かる．また，$f_9$のD-cRESは局所解に陥るものの，それ以外は多峰性関数においてDD-cRES，D-cRESとも大域最小値を発見する．

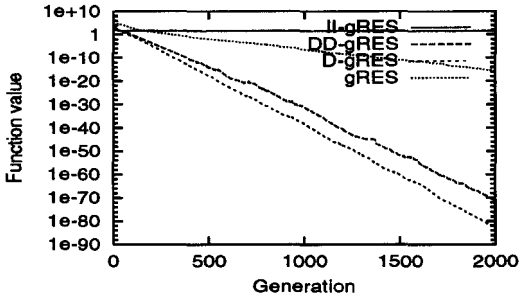
ノイズを含むテスト関数に関して最良の性能を示したMPRをTable.6.2にまとめる．これより，CESはMPRの性能は単峰性，多峰性に関係なく各問題に依存するが，FESは単峰性の関数ではDが，多峰性の関数ではDDが良い性能を示し，gRES，cRESはDが良い性能を示す傾向が見られる．



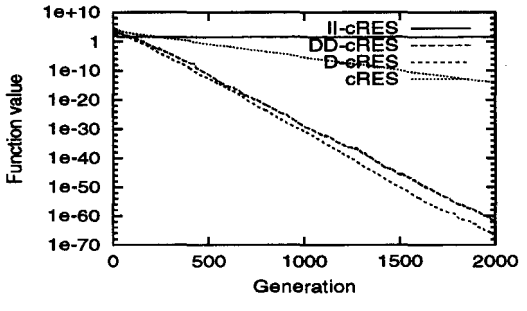

(a) CES.




(b) FES.

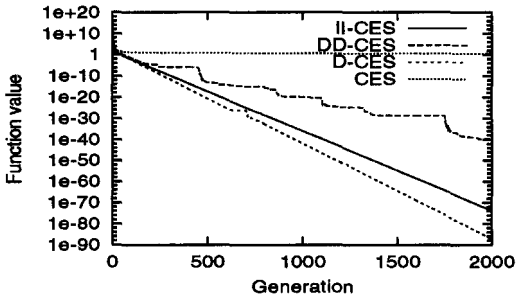



(c) gRES.

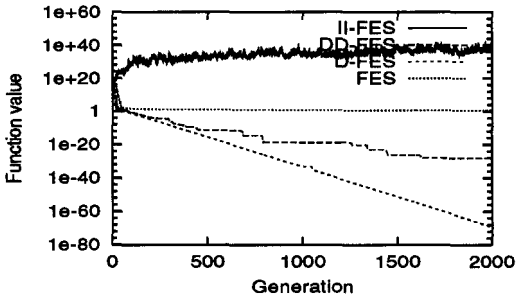



(d) cRES.

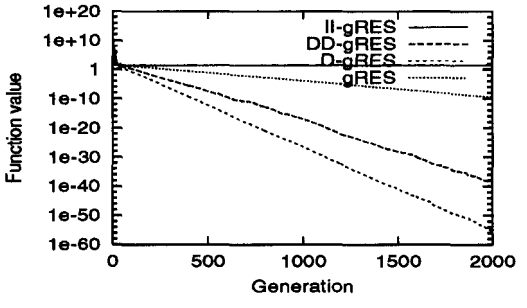
Fig. 6.1: Averaged best results on $f_1$.



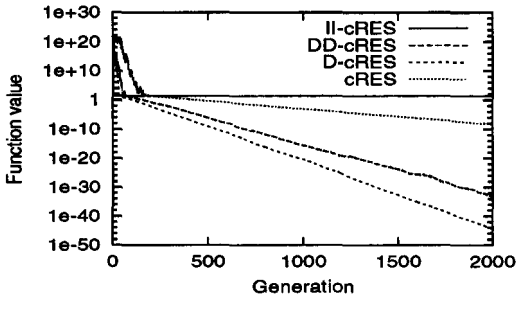

(a) CES.




(b) FES.

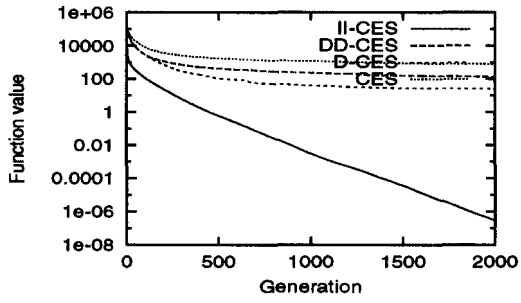



(c) gRES.

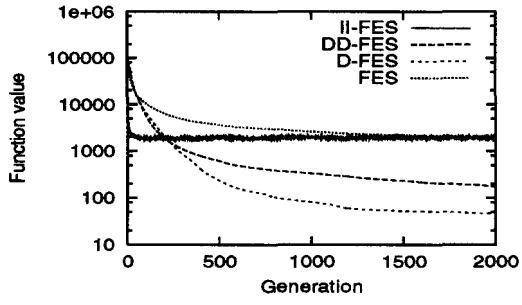



(d) cRES.

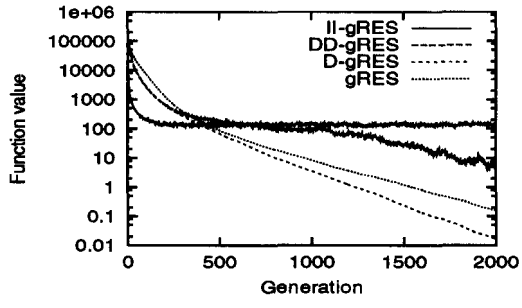
Fig. 6.2: Averaged best results on $f_2$.



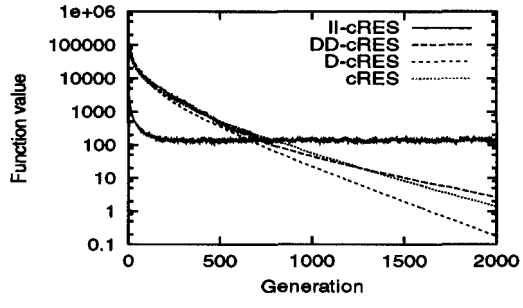

(a) CES.




(b) FES.

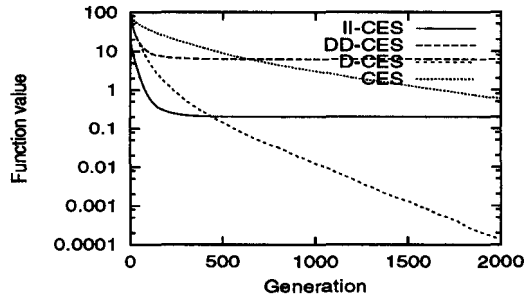



(c) gRES.

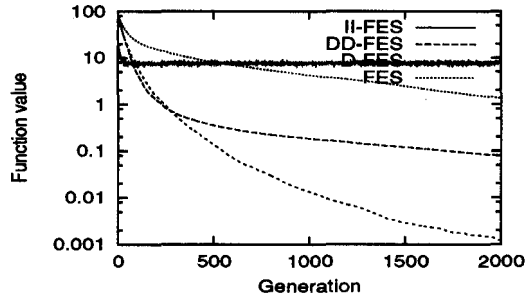



(d) cRES.

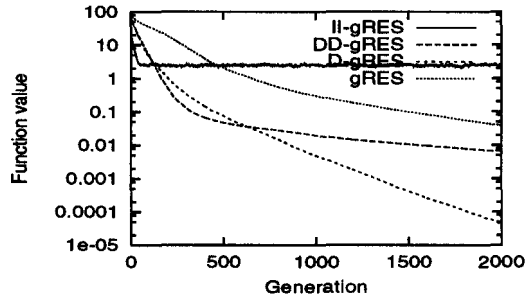
Fig. 6.3: Averaged best results on $f_3$.



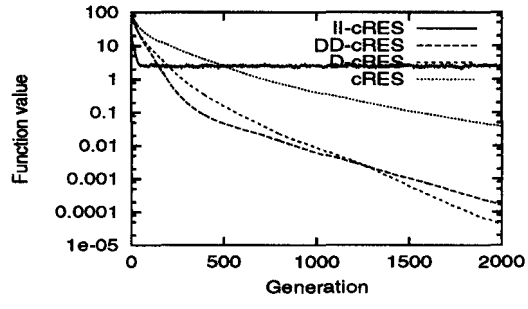

(a) CES.




(b) FES.

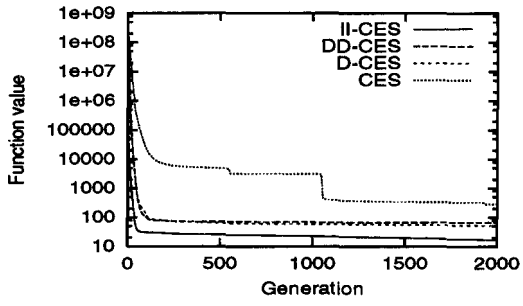



(c) gRES.

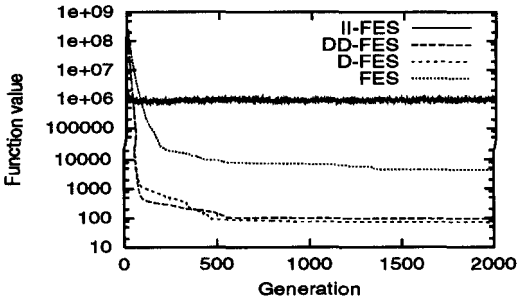



(d) cRES.

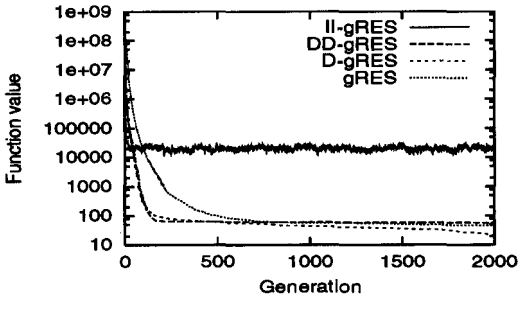
Fig. 6.4: Averaged best results on $f_4$.



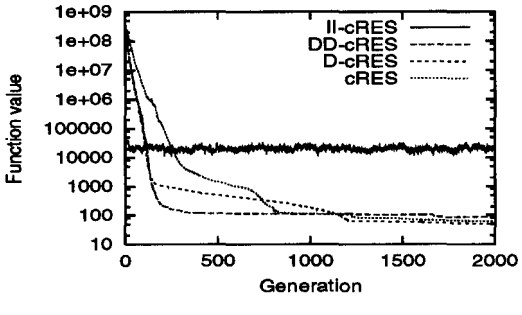

(a) CES.




(b) FES.

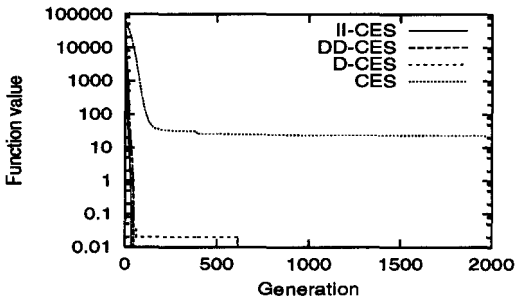



(c) gRES.

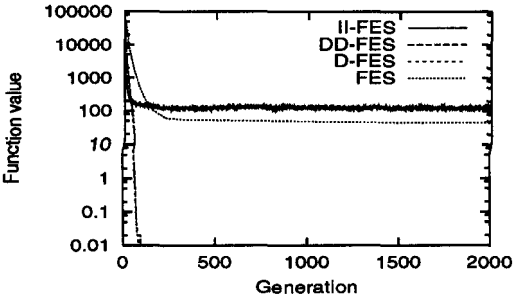



(d) cRES.

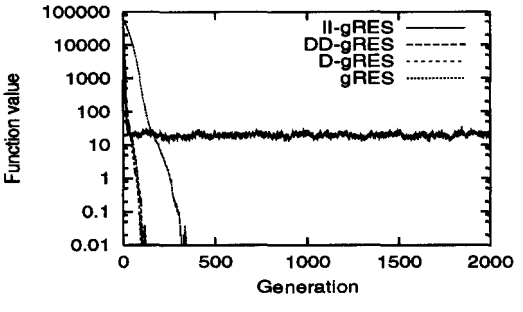
Fig. 6.5: Averaged best results on $f_5$.



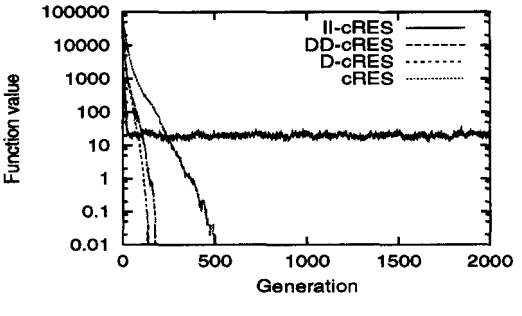

(a) CES.




(b) FES.

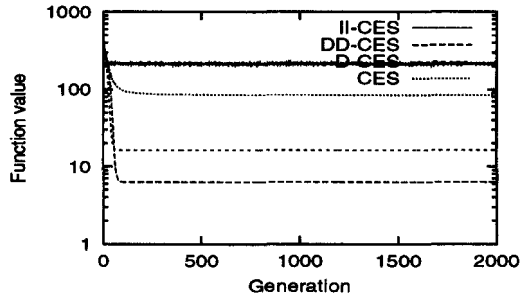



(c) gRES.

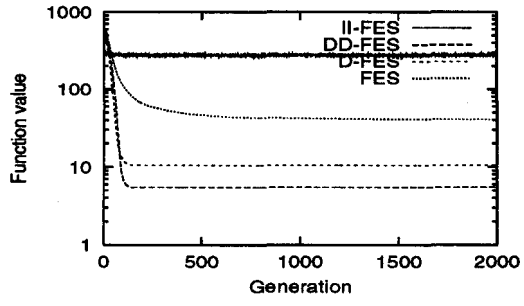



(d) cRES.

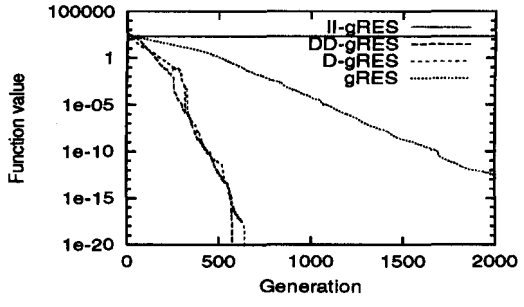
Fig. 6.6: Averaged best results on $f_6$.



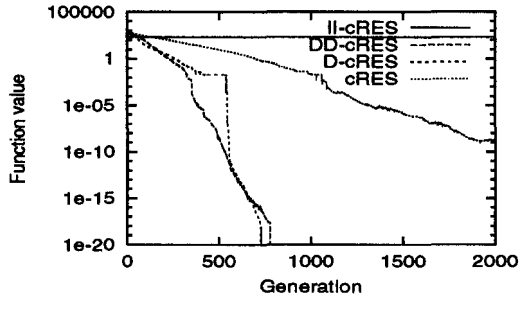

(a) CES.




(b) FES.

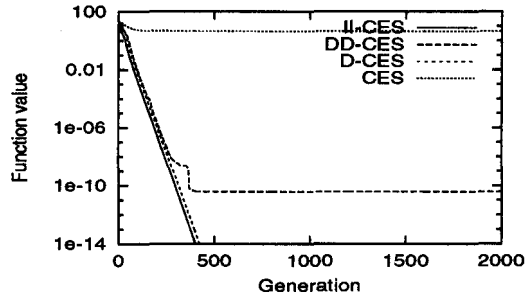



(c) gRES.

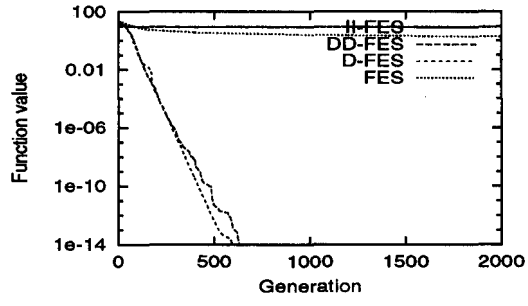



(d) cRES.

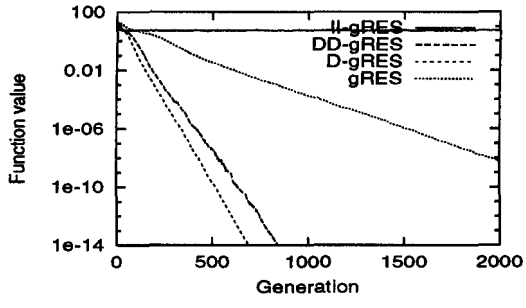
Fig. 6.7: Averaged best results on $f_7$.



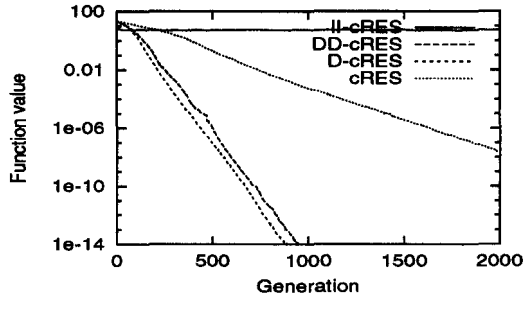

(a) CES.




(b) FES.

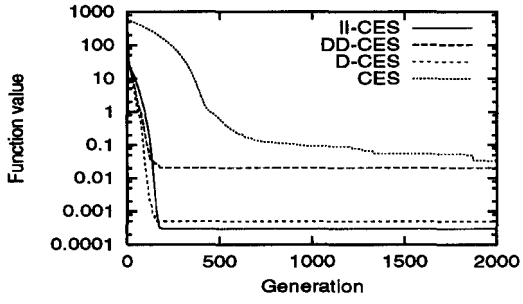



(c) gRES.

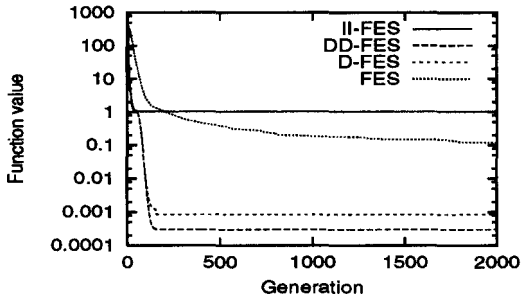



(d) cRES.

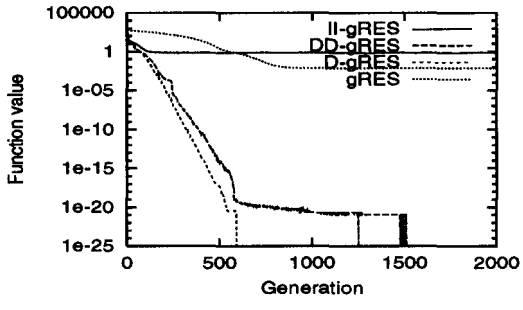
Fig. 6.8: Averaged best results on $f_8$.



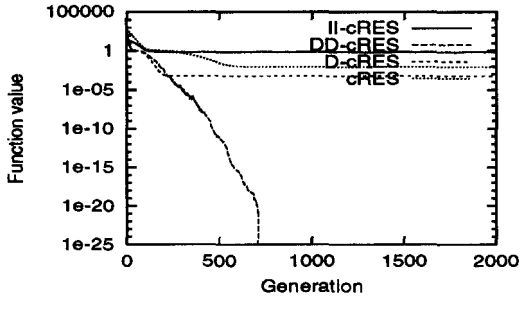

(a) CES.




(b) FES.

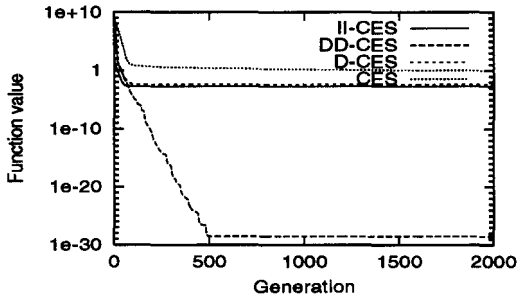



(c) gRES.

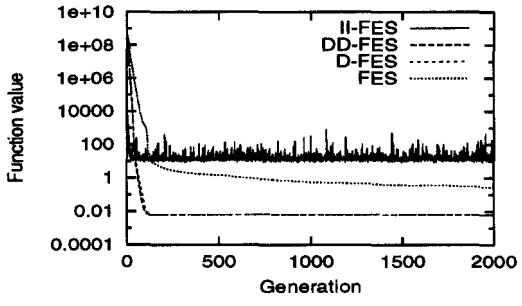



(d) cRES.

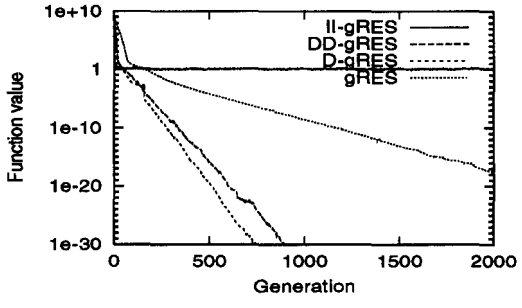
Fig. 6.9: Averaged best results on $f_9$.



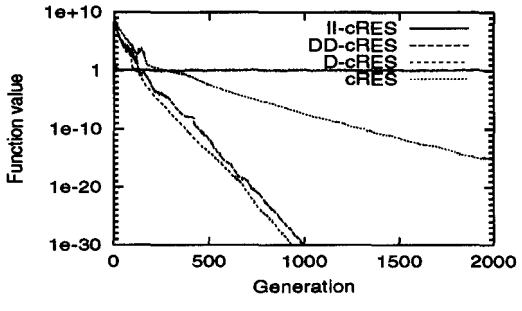

(a) CES.




(b) FES.

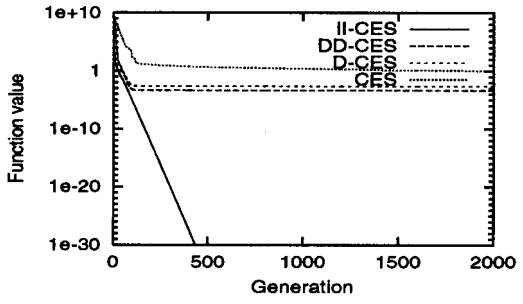



(c) gRES.

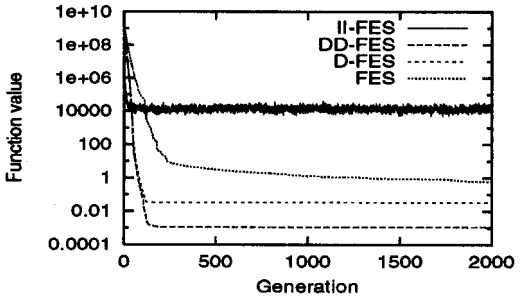



(d) cRES.

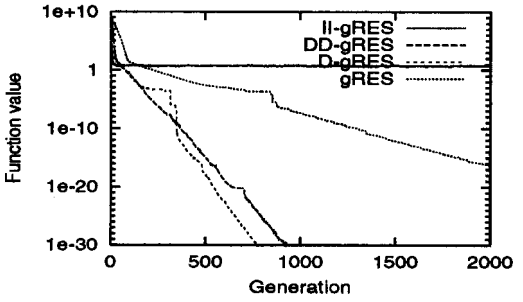
Fig. 6.10: Averaged best results on $f_{10}$.



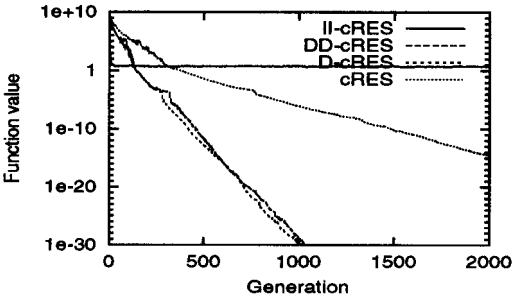

(a) CES.




(b) FES.




(c) gRES.




(d) cRES.

Fig. 6.11: Averaged best results on $f_{11}$.

### 6.2.3 進化ダイナミクス

進化過程での集団のダイナミクスを調べるため，子孫の時系列データに対して主成分分析を行う．主成分得点散布図からは集団の分布を読み取ることができ，第1主成分から第30主成分までの全ての寄与率を示した図からは集団の空間的な偏り具合が分かる．ここで取り扱うサンプルは，序盤の世代として100世代，中盤の世代として500世代，終盤の世代として1000世代とし，それぞれでの，200個体の実数値変数30次元の全18000サンプルとする．なお，Table.4.2に示す最も基本的なSphere関数を例に取り上げる．

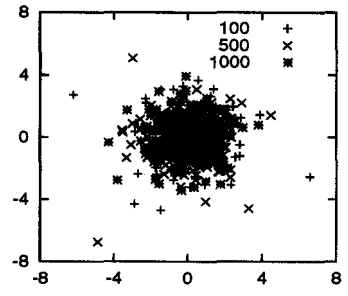
II-CES，DD-CES，D-CES，CESの主成分得点散布図をそれぞれFig. 6.12に示す．II-CES, DD-CES, D-CESは世代が進んでも広く分布しているが，CESは1000世代で一直線上に集まる．また，第1主成分から第30主成分までの全ての寄与率を示したFig. 6.13(a)(b)(c)では，II-CES, DD-CES, D-CESは全成分の寄与率が1%から6%の間に分布し，かつ進化の過程で大きな変化が見られない．これは，集団に空間的な大きな偏りは無く，その偏り具合は進化の過程で変化しないことを意味する．Fig. 6.13(d)では，CESは100世代で第2主成分までの寄与率が60%を越え，1000世代では特に第1主成分が90%を越える．これは，集団に空間的な偏りが生じていることを意味する．

II-FES，DD-FES，D-FES，FESの主成分得点散布図と全寄与率をそれぞれFig. 6.14，Fig. 6.15に示す．これらは，CESと同様の特性を示している．

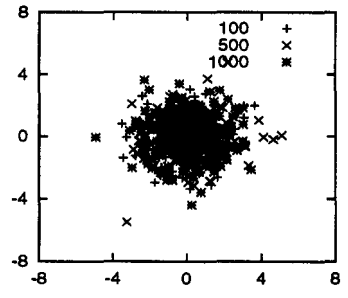
II-gRES，DD-gRES，D-gRES，gRESの主成分得点散布図をそれぞれFig. 6.16に示す．II-gRES, DD-gRES, D-gRES，gRESは世代が進んでも広く分布している．また，全寄与率を示したFig. 6.17(a)では，II-gRESは寄与率が1%から6%の間に分布し，かつ進化の過程で大きな変化が見られない．Fig. 6.17(b)(c)では，DD-gRES, D-gRESは1000世代で約20主成分程度が3.33%の寄与率である．これは，実数値変数が30次元であるため，進化の過程で空間的に限りなく偏りが無くなるように移り変わることを意味している．Fig. 6.17(d)では，gRESは100世代で第2主成分までの寄与率が30%を越え，1000世代で約10主成分程度が3.33%の寄与率である．DD-gRES，D-gRESは1000世代で約20主成分程度であるため，gRESはDDとDを用いることによって早い世代で極端に偏りが無くなることが分かる．

II-cRES，DD-cRES，D-cRES，cRESの主成分得点散布図と全寄与率をそれぞれFig. 6.18，Fig. 6.17に示す．これらは，gRESと同様の特性を示している．

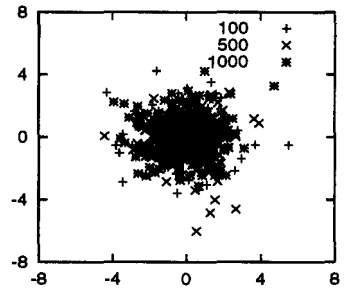
なお，ここでは示さないが，他の関数に関しても，CES，FES，gRES，cRESはSphere関数と同様の特性を示した．



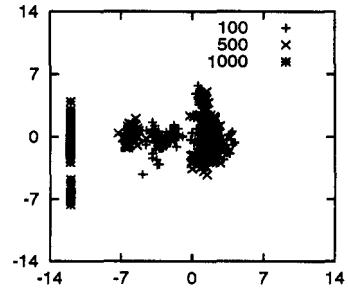

(a) II-CES.




(b) DD-CES.

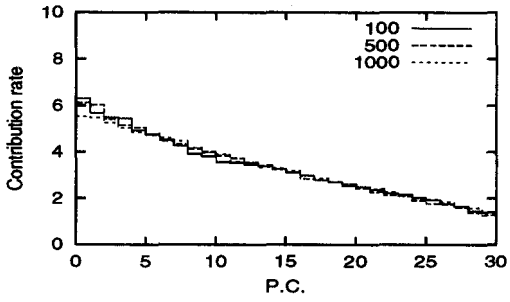



(c) D-CES.

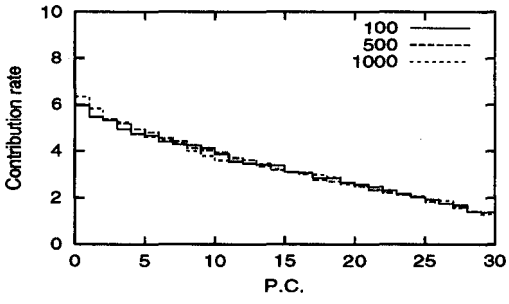



(d) CES.

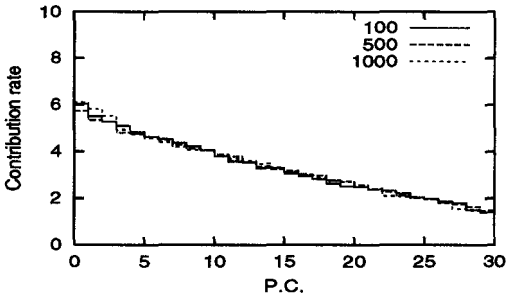
Fig. 6.12: Principal components on CES for $f_1$ when the generation is 100, 500 and 1000.



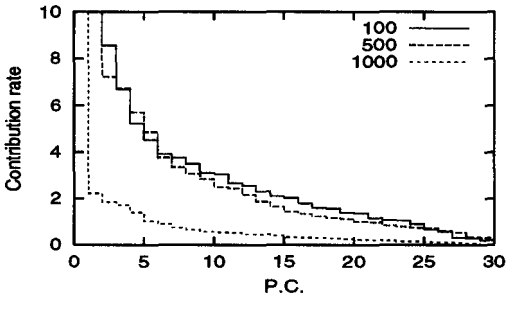

(a) II-CES.




(b) DD-CES.

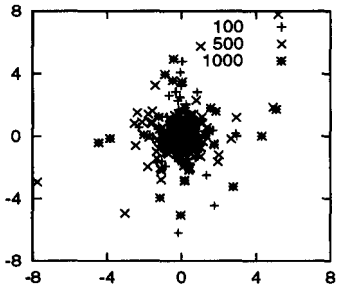



(c) D-CES.

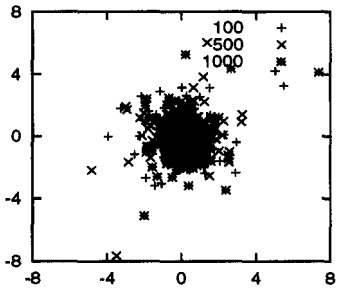



(d) CES.

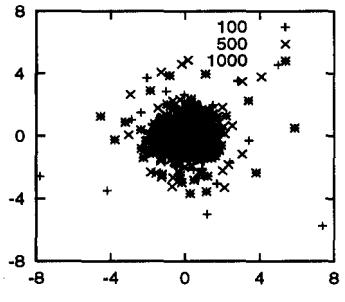
Fig. 6.13: Contribution rates on CES for $f_1$ when the generation is 100, 500 and 1000.



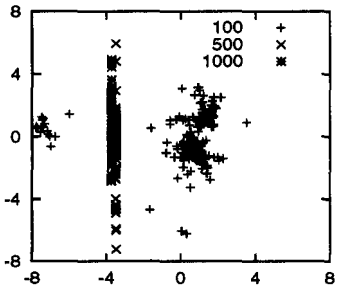

(a) II-FES.




(b) DD-FES.

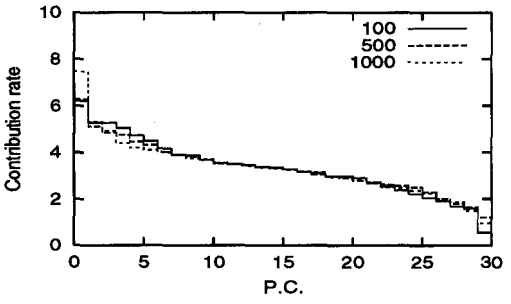



(c) D-FES.

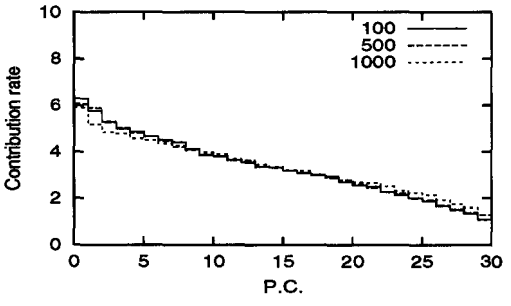



(d) FES.

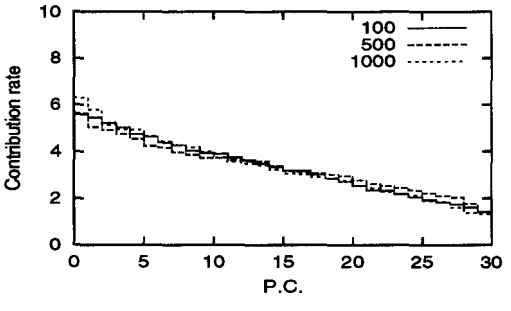
Fig. 6.14: Principal components on FES for $f_1$ when the generation is 100, 500 and 1000.



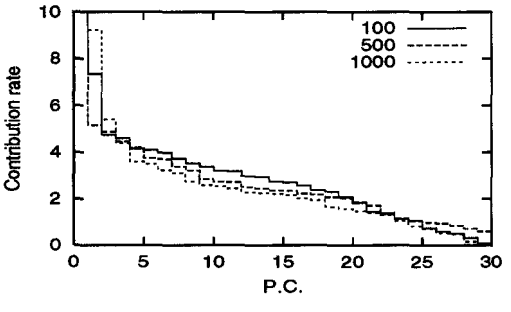

(a) II-FES.




(b) DD-FES.

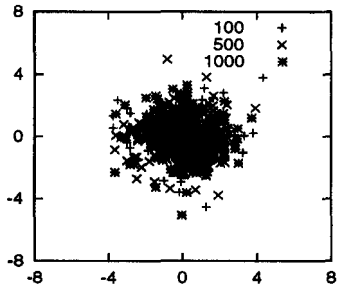



(c) D-FES.

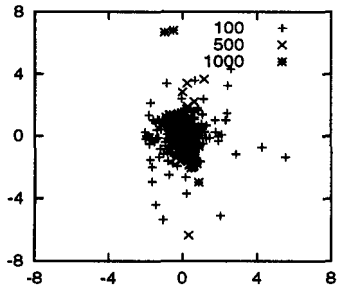



(d) FES.

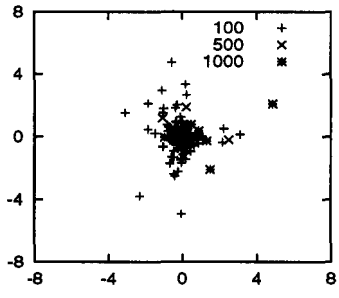
Fig. 6.15: Contribution rates on FES for $f_1$ when the generation is 100, 500 and 1000.



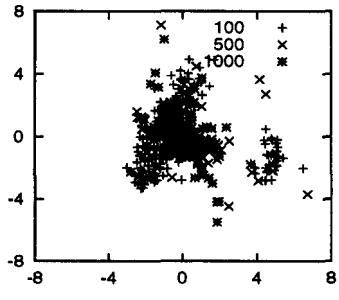

(a) II-gRES.




(b) DD-gRES.

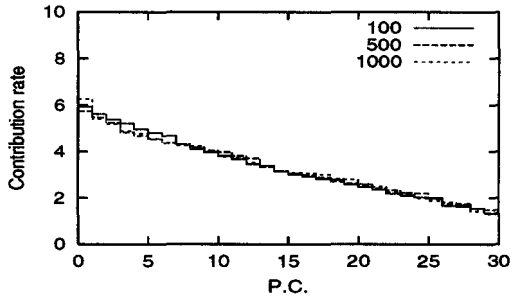



(c) D-gRES.

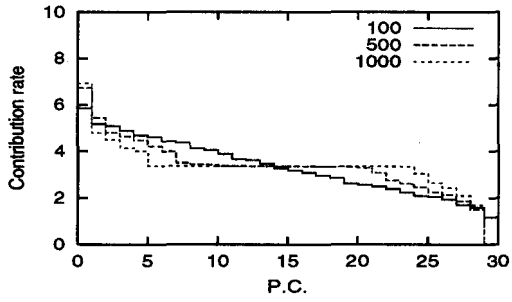



(d) gRES.

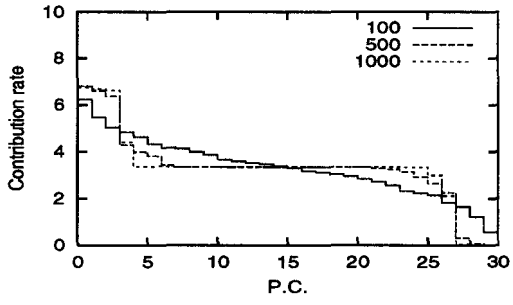
Fig. 6.16: Principal components on gRES for $f_1$ when the generation is 100, 500 and 1000.



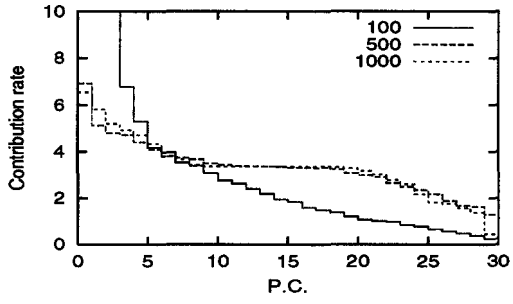

(a) II-gRES.




(b) DD-gRES.

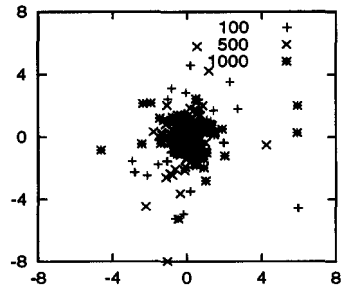



(c) D-gRES.

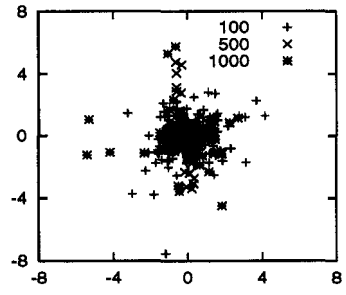



(d) gRES.

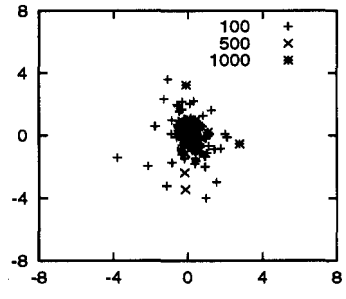
Fig. 6.17: Contribution rates on gRES for $f_1$ when the generation is 100, 500 and 1000.



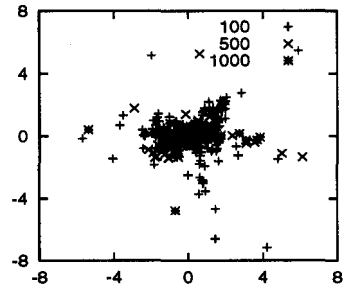

(a) II-cRES.




(b) DD-cRES.

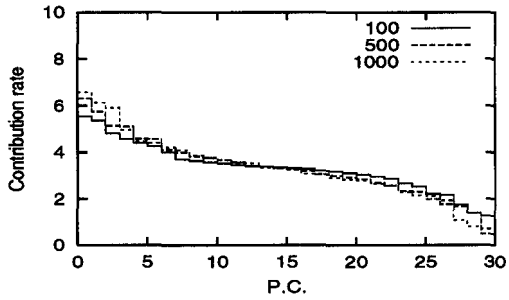



(c) D-cRES.

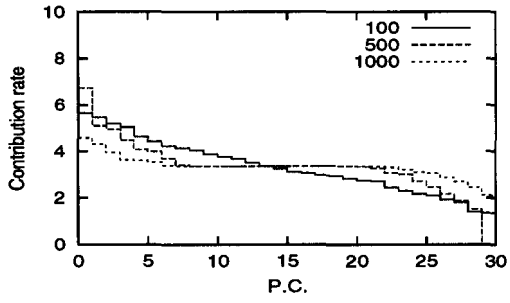



(d) cRES.

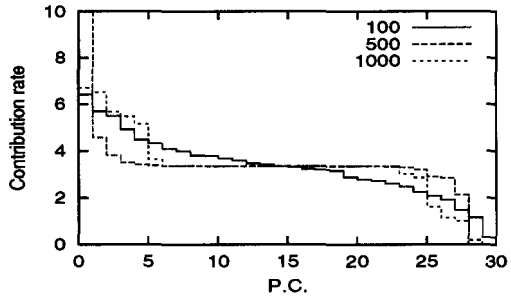
Fig. 6.18: Principal components on cRES for $f_1$ when the generation is 100, 500 and 1000.



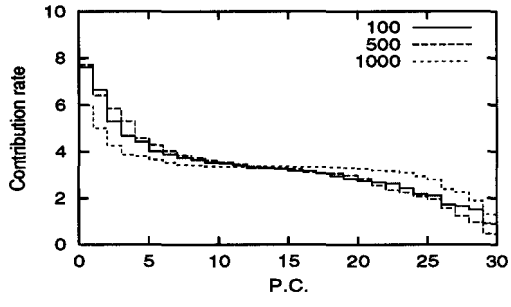

(a) II-cRES.




(b) DD-cRES.




(c) D-cRES.




(d) cRES.

Fig. 6.19: Contribution rates on cRES for $f_1$ when the generation is 100, 500 and 1000.

### 6.2.4 探索特徴 (まとめ)

以上より，CES，FES，gRES，cRES の探索特徴を以下のようにまとめることができる．

- CES の II は性能が向上しない場合もあるが，DD，D は性能が向上する．CES に MPR を用いた場合，集団は空間的な偏りを生じることなく分布し，その偏り具合は世代を通じて変化しない．
- FES は DD，D を用いることにより性能は向上する．FES に MPR を用いた場合，CES と同様の進化ダイナミクスを示す．
- gRES，cRES は DD，D を用いることにより性能は向上する．gRES，cRES に MPR を用いた場合，集団は空間的な偏りを生じることなく分布している．特に DD，D を用いる場合は，早い世代で限りなく偏りのない分布に移り変わる．

## 6.3 ノイズを含むテスト関数を用いた計算機実験

工学問題に応用する場合に重要となるMPRのノイズに対する頑健性を明らかにするため，Bäck と Hammel [12][58] が用いた以下のノイズを含むテスト関数を用いて計算機実験を行う:

$$F(\boldsymbol{x}_i) = f(\boldsymbol{x}_i) + \sigma_\delta N_i(0,1) \tag{6.1}$$

ここで，$f(\boldsymbol{x}_i)$ として，Table.4.2に示す関数を例に取り上げる．なお，$\sigma_\delta$ を，0.0, 0.001, 0.01, 0.1, 1.0 の5種類の値に設定して計算機実験を行った．$f_1$ から $f_6$ までは単峰性，$f_7$ から $f_{11}$ は多峰性関数である．すべてのテスト関数は30次元の探索空間を持ち，大域最小値は0である．

CES，FESおよびgRES，cRESは，Yao と Liu [117][119] に従って，$(\mu, \lambda) = (30, 200)$，相関突然変異を用いないこととした．戦略パラメータの上限 $\eta_{max}$ は探索範囲が比較的狭い $f_7$ のみ 1.0，その他では3.0とし，それぞれ50回試行を繰り返した．さらに，gRESと cRESは，$m = 5$ として，$g_{dup}$，$g_{del}$，$g_{inv}$ の適用確率 (P) を，それぞれ，$P(g_{dup}) = 0.6$，$P(g_{del}) = 0.3$，$P(g_{inv}) = 0.1$ とした [79][82].

なお，これらの推奨パラメータ値は，予備的な実験結果から決定した．また，同実験から，パラメータ値の変化に関してRESは非常に頑健であることが分かったため，パラメータ値チューニングにはそれほど注意を払っていない．それゆえ，詳細な実験を通してこれらを決定すれば，パフォーマンスは更に向上すると思われる．

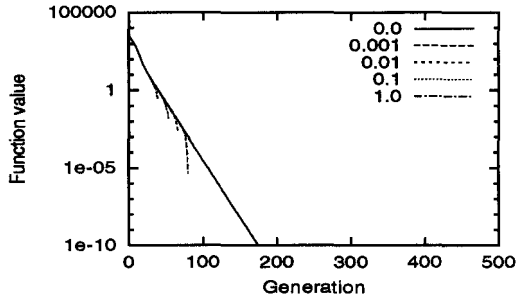
テスト関数 $f_1$ のCES，FES，gRES，cRES 50試行の平均結果をそれぞれFig. 6.20，Fig. 6.21，Fig. 6.22，Fig. 6.23 に示す．縦軸には最良個体の関数値の50回平均を，横軸には世代をとる．

Fig. 6.20 で，CESはノイズの影響を受けるが，II-CES，DD-CES，D-CESは，計算機実験で用いた $\sigma_\delta$ のノイズの影響はほとんど見受けられない．同様に，Fig. 6.21 のFESで，Fig. 6.22 のgRESで，Fig. 6.23 のcRESで，DD，Dの MPR を用いた場合，ノイズの影響をほとんど受けない．

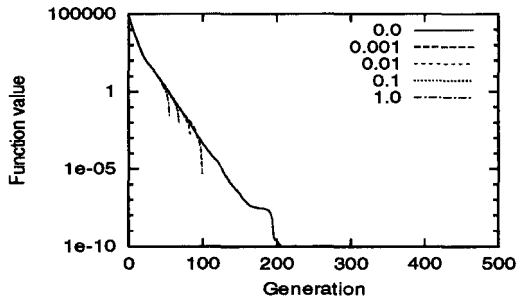
他のテスト関数に関して得られた結果をTable.6.3にまとめる．なお，○はMPRを用いない場合に比べてノイズの影響を受けず性能が向上することを意味し，△はMPRを用いない場合に比べて性能は向上するがノイズの影響を受けることを意味し，×はノイズの影響を受け，性能が向上しないことを意味する．ノイズを含むテスト関数に対して，CESはMPR を用いた場合はほぼ性能が向上することが見られるが，どのMPRを用いれば良いかは問題に依存する．なお，Dを用いれば全てのテスト関数において性能が向上する．また，FESはDD,Dを用いれば性能が向上し，gRES，cRESはDを用いれば全てのテスト関数において性能が向上する．

Table. 6.3: Summary of experiments for noisy test functions.

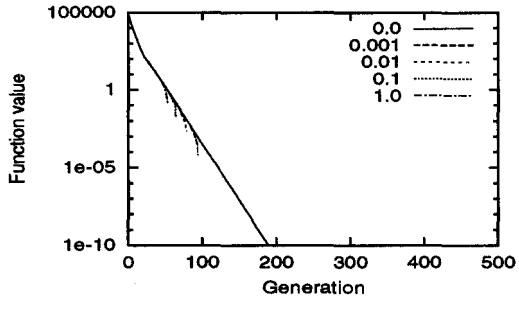
| | | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ | $f_8$ | $f_9$ | $f_{10}$ | $f_{11}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CES | II | ○ | ○ | △ | △ | △ | ○ | × | △ | △ | △ | ○ |
| | DD | ○ | ○ | △ | × | △ | ○ | △ | ○ | △ | ○ | △ |
| | D | ○ | ○ | △ | △ | △ | △ | △ | ○ | △ | △ | △ |
| FES | II | × | × | × | × | × | × | × | × | × | × | × |
| | DD | ○ | ○ | △ | △ | △ | ○ | △ | △ | △ | △ | △ |
| | D | ○ | ○ | △ | △ | △ | ○ | △ | △ | △ | △ | △ |
| gRES | II | × | × | × | × | × | × | × | × | × | × | × |
| | DD | ○ | ○ | × | △ | △ | ○ | ○ | ○ | △ | ○ | ○ |
| | D | ○ | ○ | △ | △ | △ | ○ | ○ | ○ | △ | ○ | ○ |
| cRES | II | × | × | × | × | × | × | × | × | × | × | × |
| | DD | ○ | ○ | × | △ | △ | ○ | ○ | △ | △ | ○ | ○ |
| | D | ○ | ○ | △ | △ | △ | ○ | △ | △ | △ | ○ | ○ |



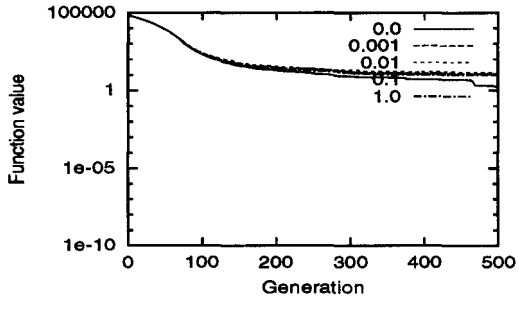

(a) II-CES.




(b) DD-CES.

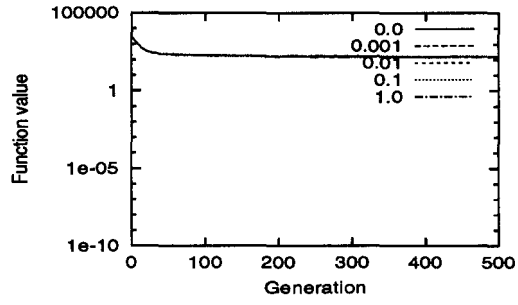



(c) D-CES.

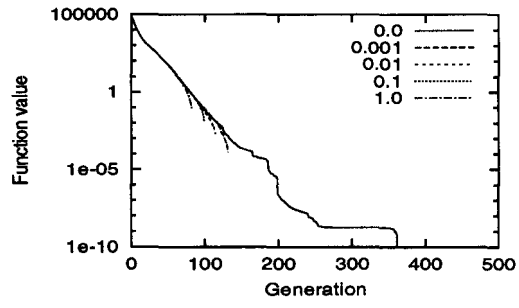



(d) CES.

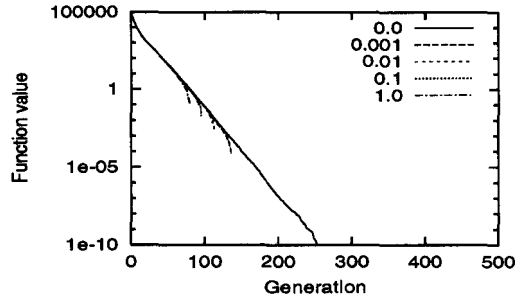
Fig. 6.20: Averaged best results on CES for $f_1$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



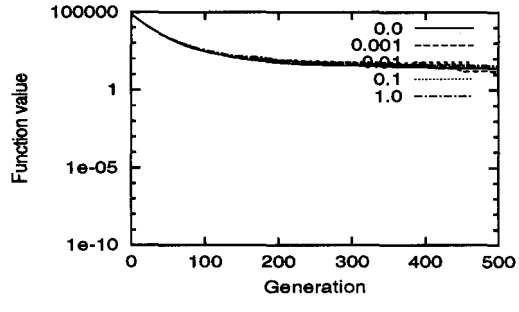

(a) II-FES.




(b) DD-FES.

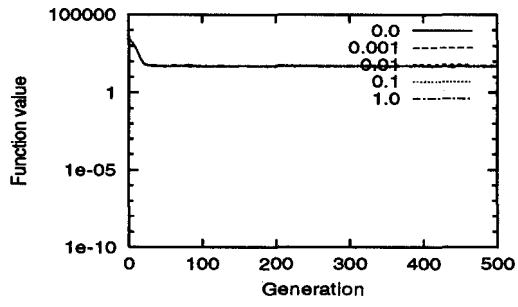



(c) D-FES.

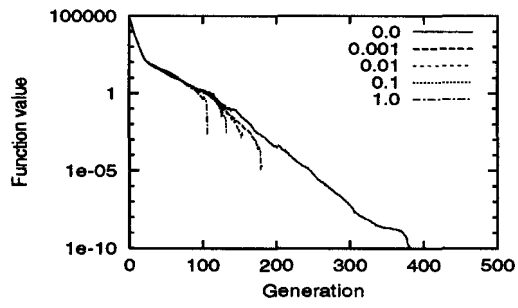



(d) FES.

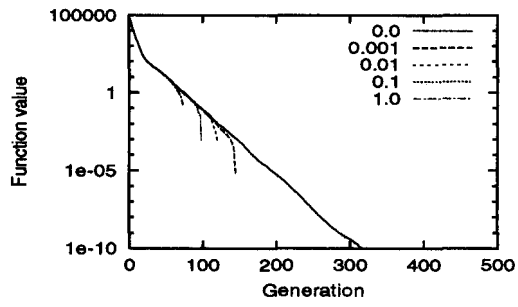
Fig. 6.21: Averaged best results on FES for $f_1$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



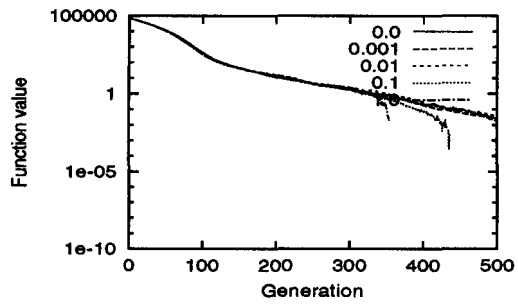

(a) II-gRES.




(b) DD-gRES.

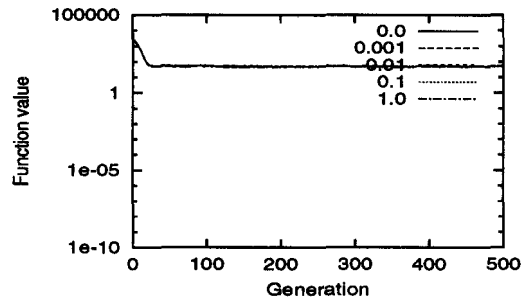



(c) D-gRES.

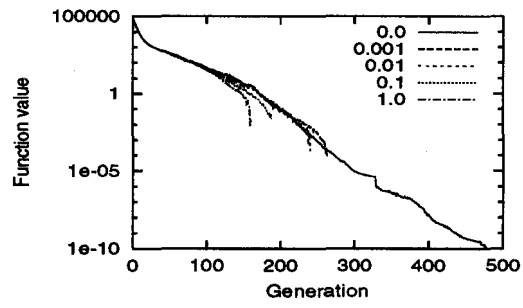



(d) gRES.

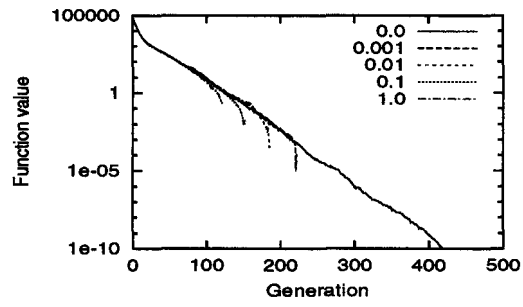
Fig. 6.22: Averaged best results on gRES for $f_1$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



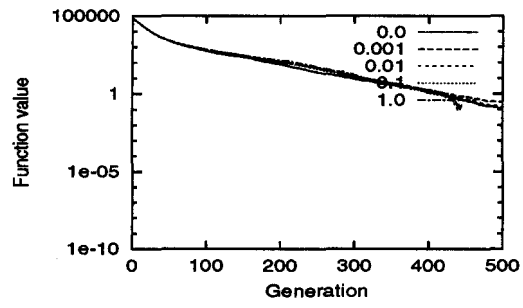

(a) II-cRES.




(b) DD-cRES.

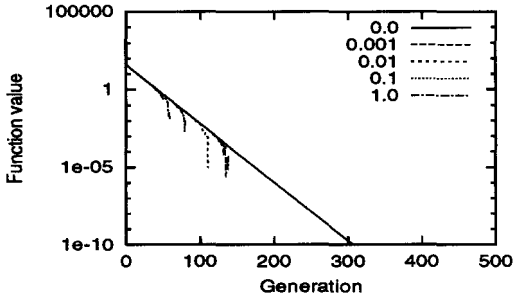



(c) D-cRES.

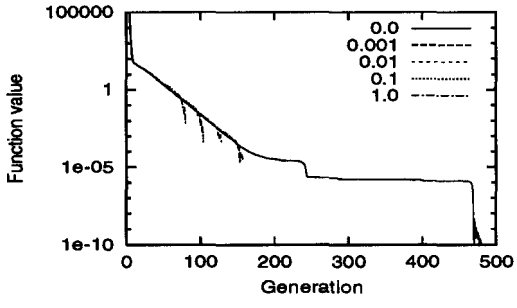



(d) cRES.

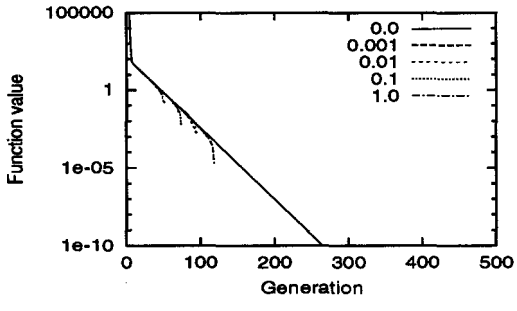
Fig. 6.23: Averaged best results on cRES for $f_1$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



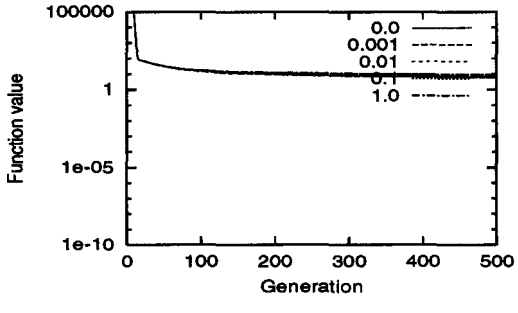

(a) II-CES.




(b) DD-CES.

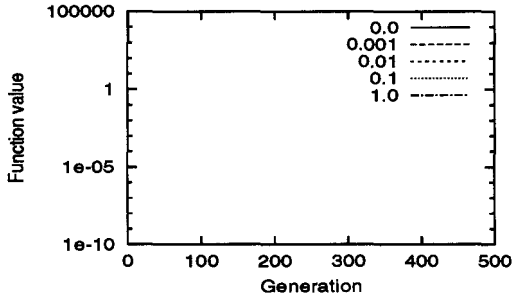



(c) D-CES.

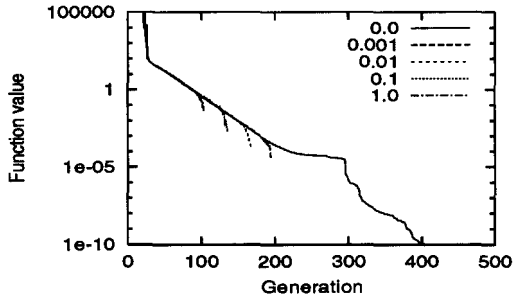



(d) CES.

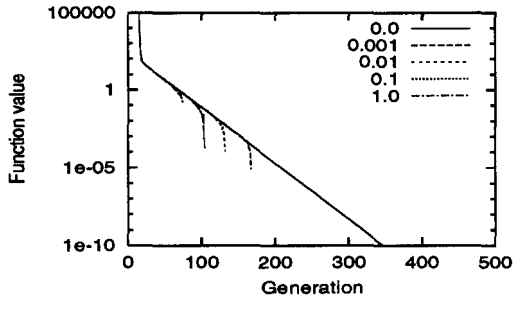
Fig. 6.24: Averaged best results on CES for $f_2$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



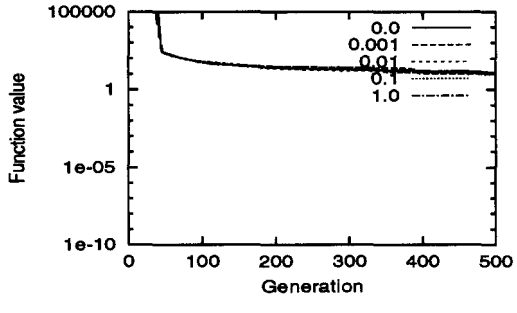

(a) II-FES.




(b) DD-FES.

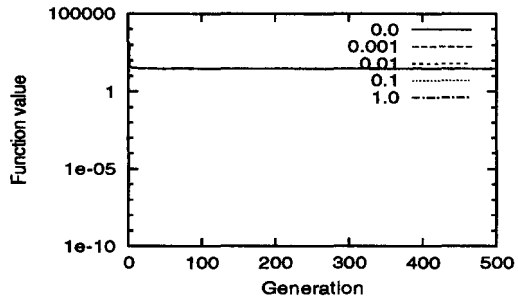



(c) D-FES.

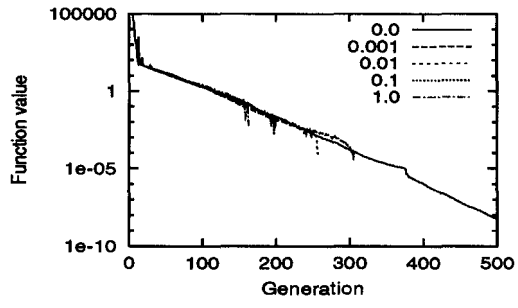



(d) FES.

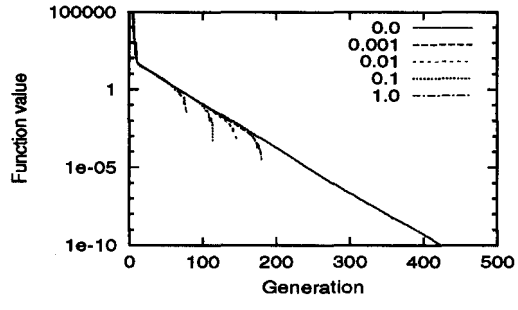
Fig. 6.25: Averaged best results on FES for $f_2$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



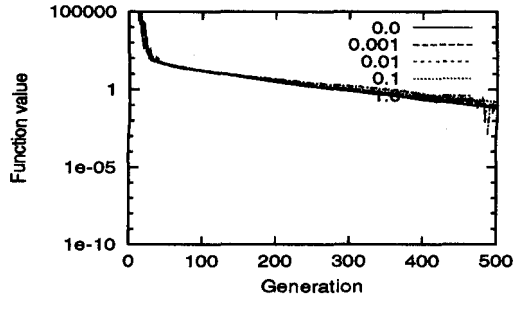

(a) II-gRES.




(b) DD-gRES.

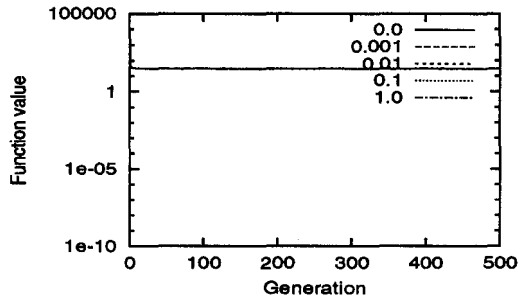



(c) D-gRES.

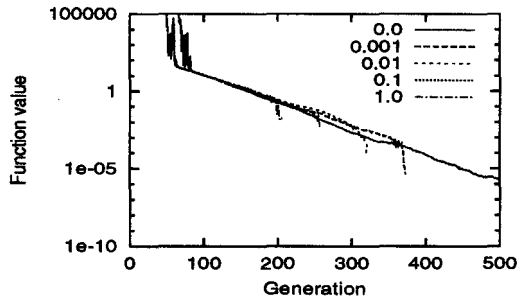



(d) gRES.

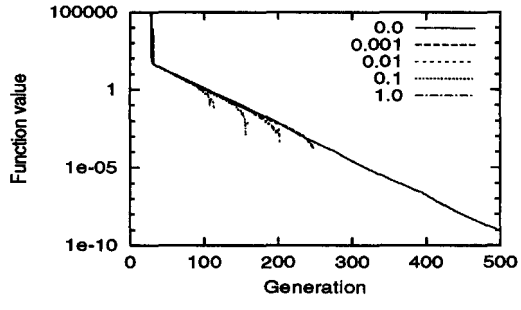
Fig. 6.26: Averaged best results on gRES for $f_2$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



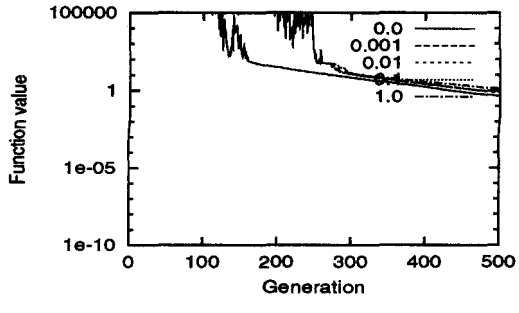

(a) II-cRES.




(b) DD-cRES.

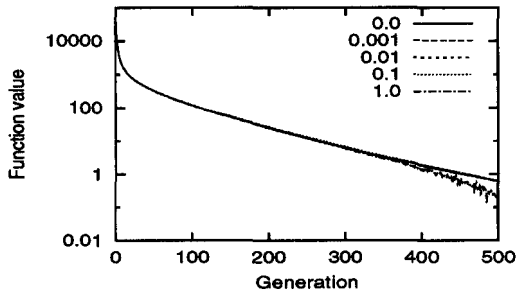



(c) D-cRES.

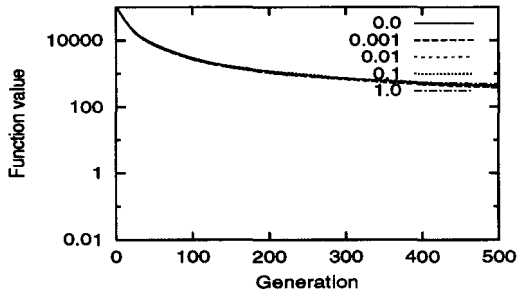



(d) cRES.

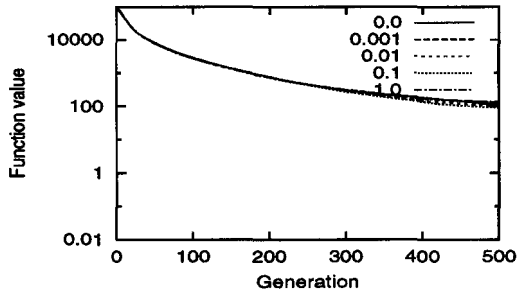
Fig. 6.27: Averaged best results on cRES for $f_2$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



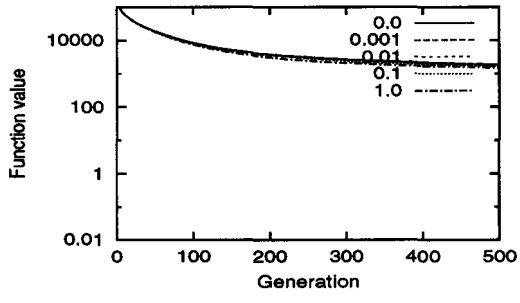

(a) II-CES.




(b) DD-CES.

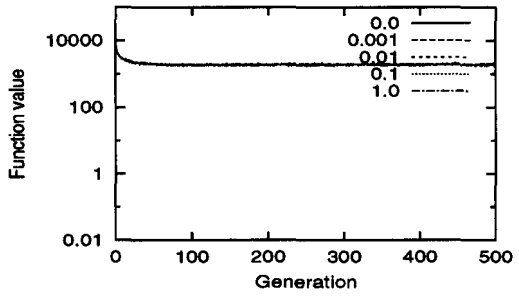



(c) D-CES.

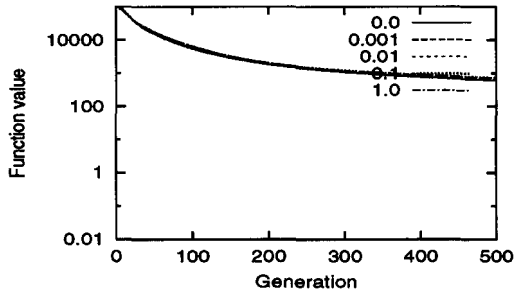



(d) CES.

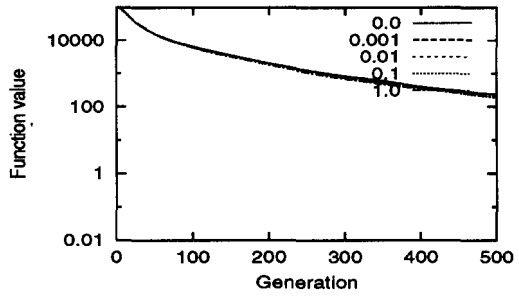
Fig. 6.28: Averaged best results on CES for $f_3$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



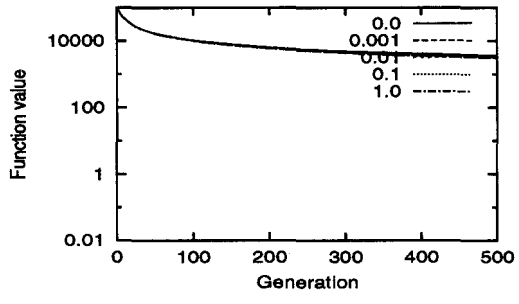

(a) II-FES.




(b) DD-FES.

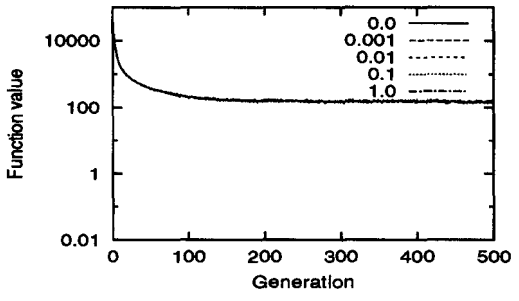



(c) D-FES.

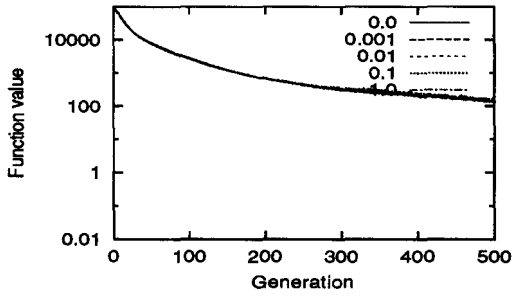



(d) FES.

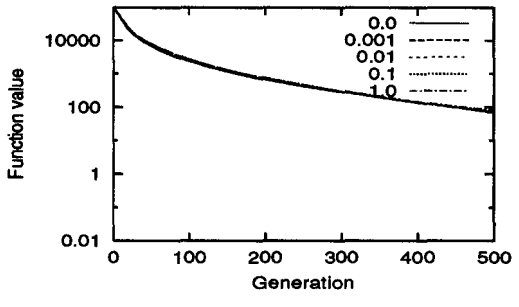
Fig. 6.29: Averaged best results on FES for $f_3$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



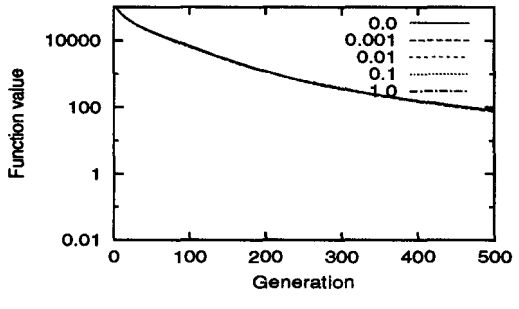

(a) II-gRES.




(b) DD-gRES.

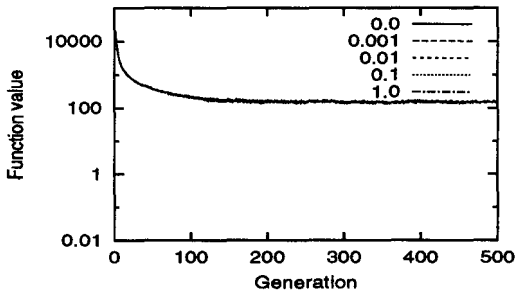



(c) D-gRES.

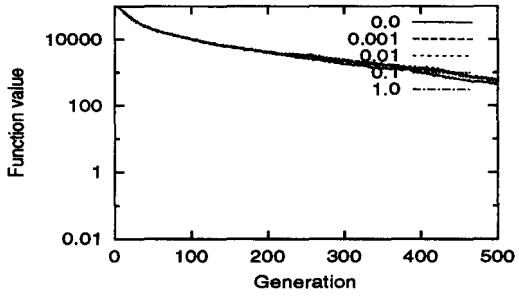



(d) gRES.

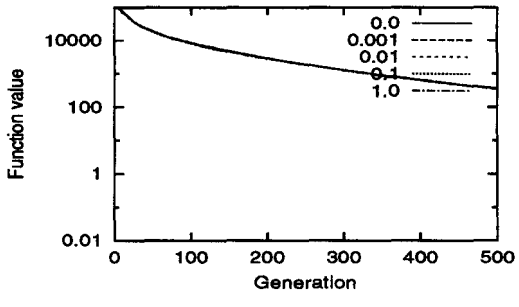
Fig. 6.30: Averaged best results on gRES for $f_3$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



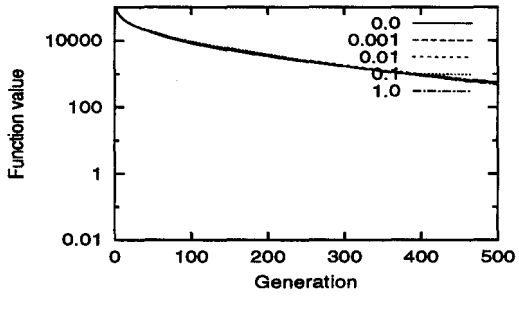

(a) II-cRES.




(b) DD-cRES.

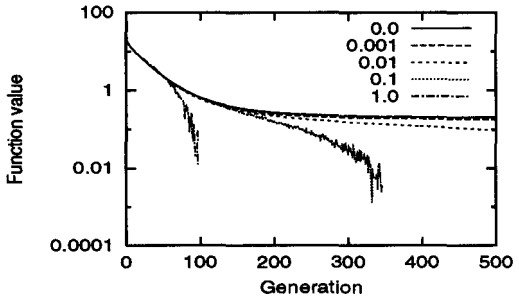



(c) D-cRES.

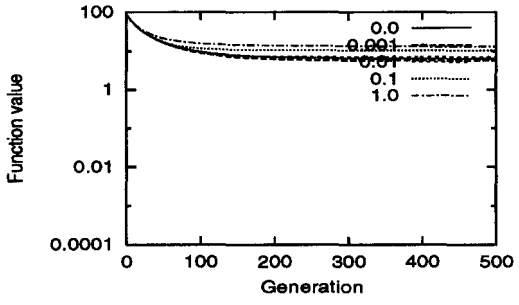



(d) cRES.

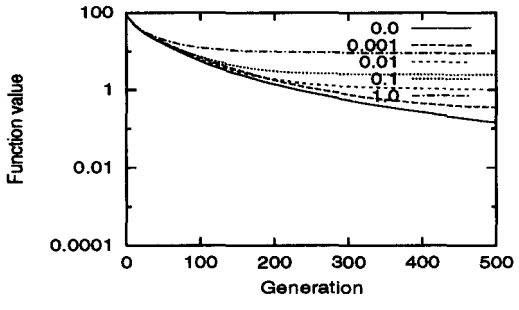
Fig. 6.31: Averaged best results on cRES for $f_3$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



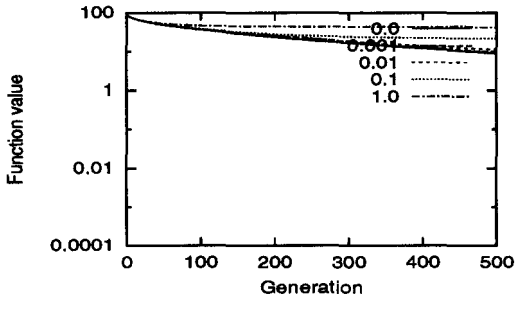

(a) II-CES.




(b) DD-CES.

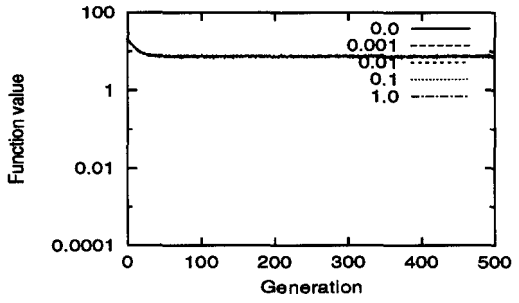



(c) D-CES.

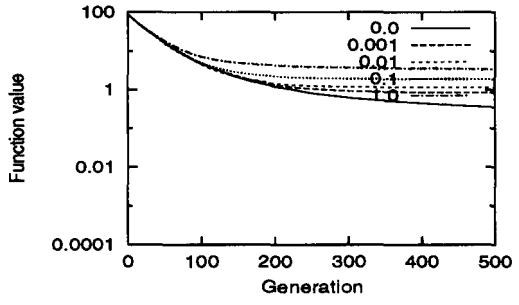



(d) CES.

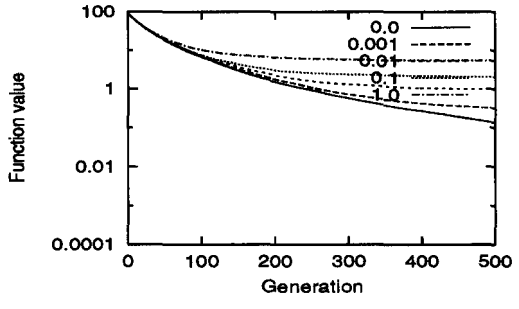
Fig. 6.32: Averaged best results on CES for $f_4$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



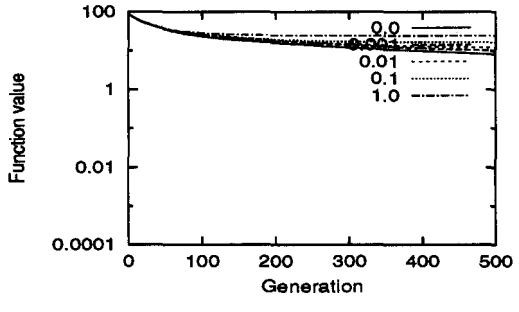

(a) II-FES.




(b) DD-FES.

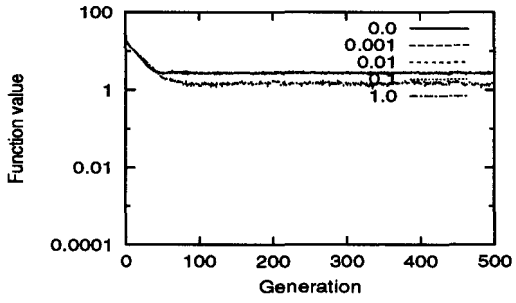



(c) D-FES.

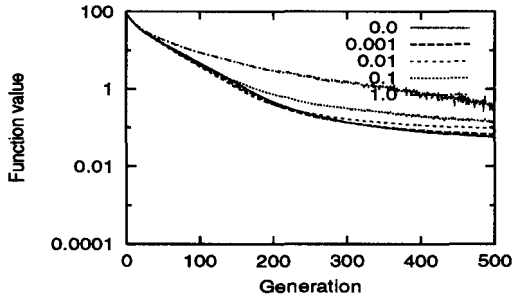



(d) FES.

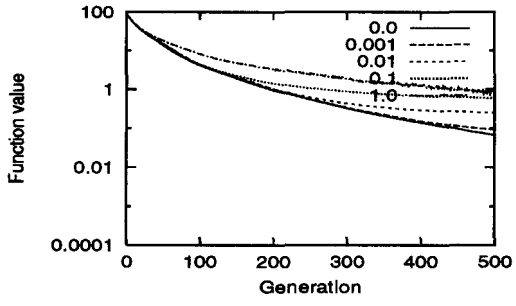
Fig. 6.33: Averaged best results on FES for $f_4$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



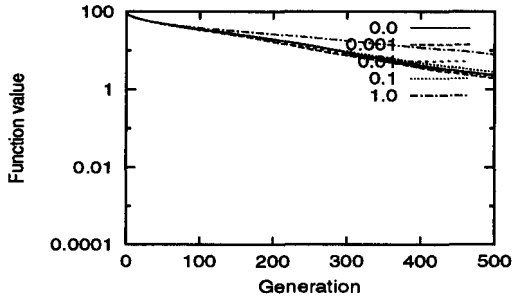

(a) II-gRES.




(b) DD-gRES.

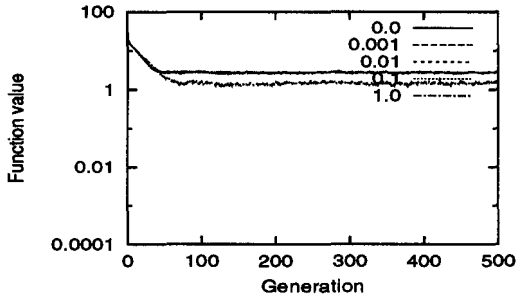



(c) D-gRES.

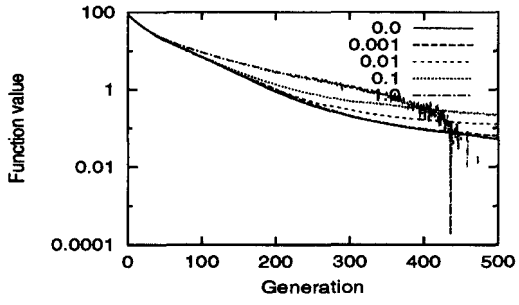



(d) gRES.

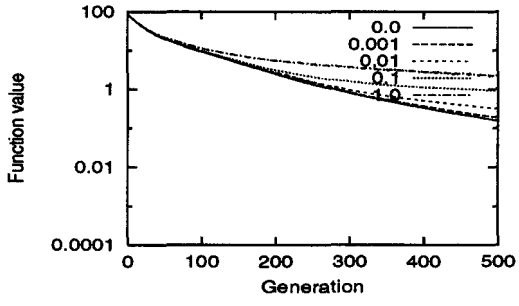
Fig. 6.34: Averaged best results on gRES for $f_4$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



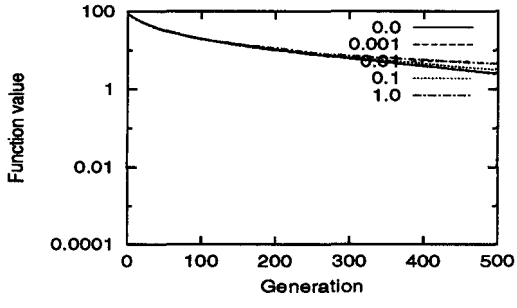

(a) II-cRES.




(b) DD-cRES.

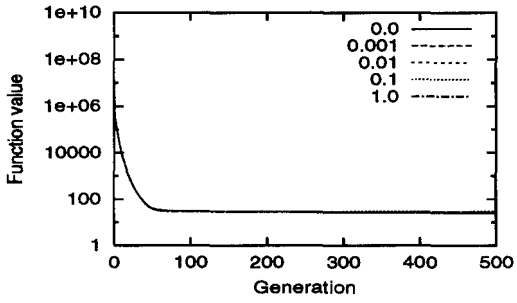



(c) D-cRES.

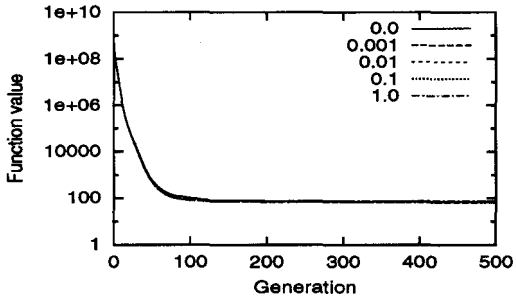



(d) cRES.

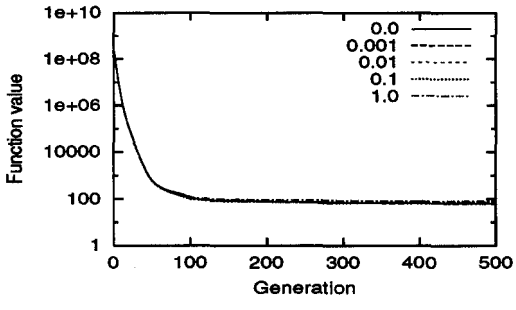
Fig. 6.35: Averaged best results on cRES for $f_4$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



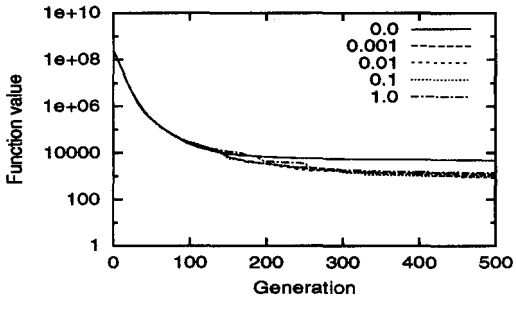

(a) II-CES.




(b) DD-CES.

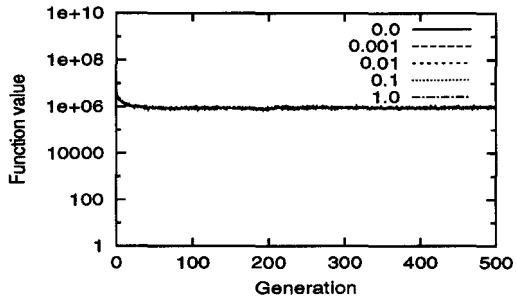



(c) D-CES.

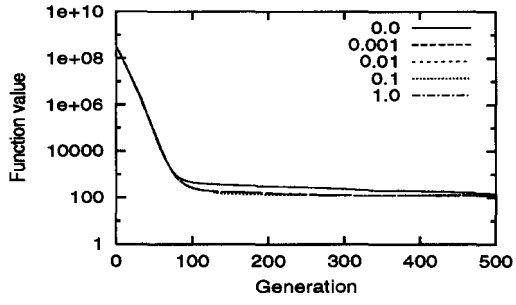



(d) CES.

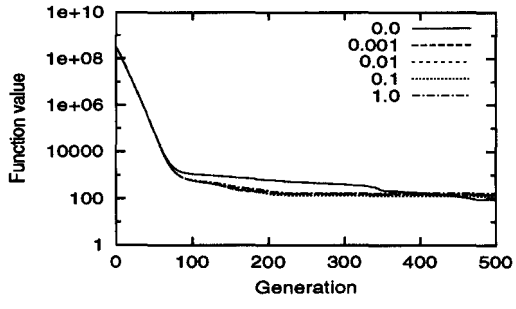
Fig. 6.36: Averaged best results on CES for $f_5$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



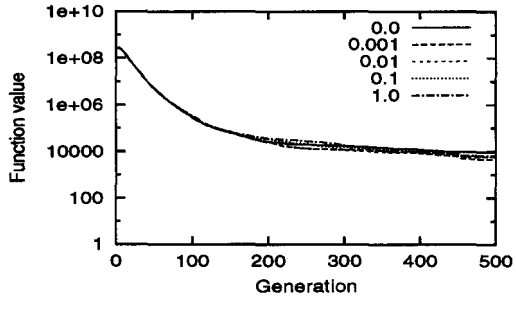

(a) II-FES.




(b) DD-FES.

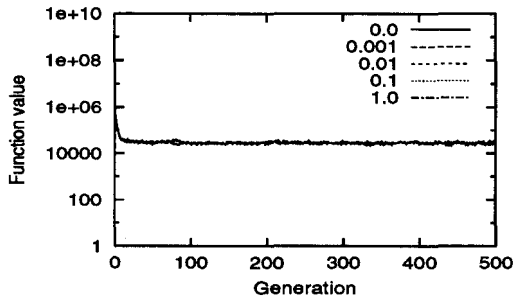



(c) D-FES.

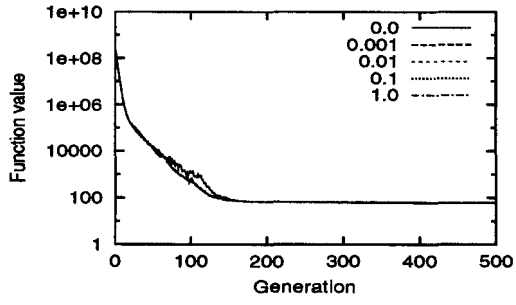



(d) FES.

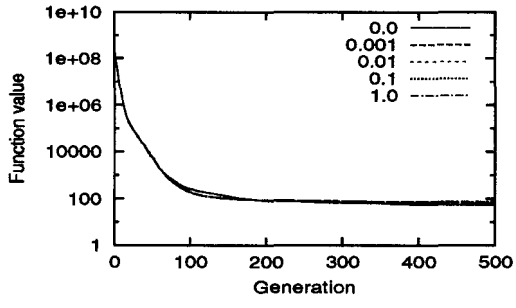
Fig. 6.37: Averaged best results on FES for $f_5$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



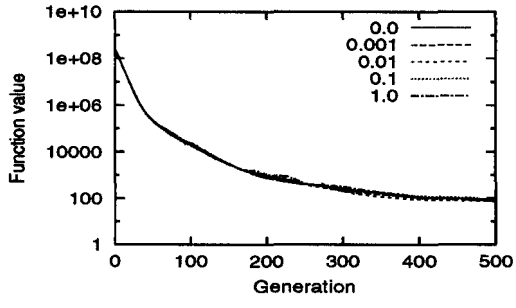

(a) II-gRES.




(b) DD-gRES.

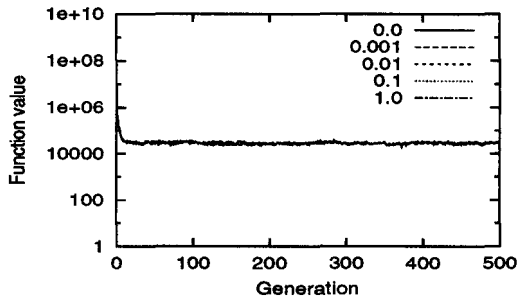



(c) D-gRES.

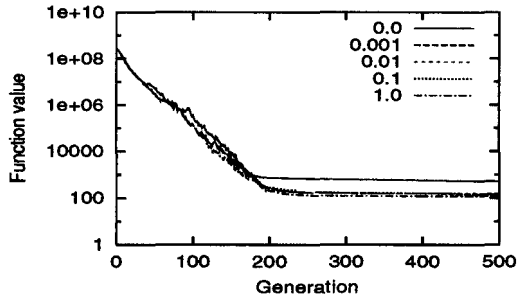



(d) gRES.

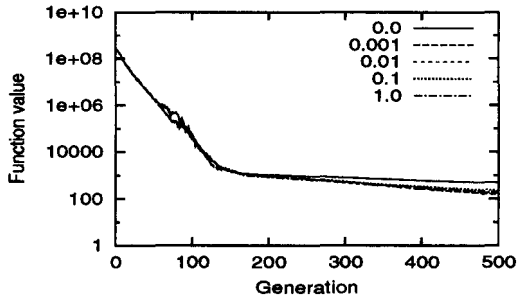
Fig. 6.38: Averaged best results on gRES for $f_5$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



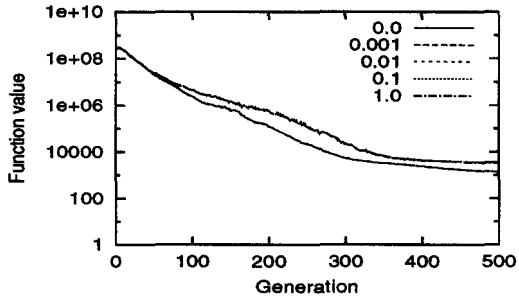

(a) II-cRES.




(b) DD-cRES.

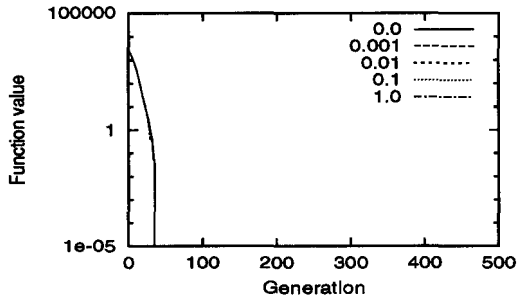



(c) D-cRES.

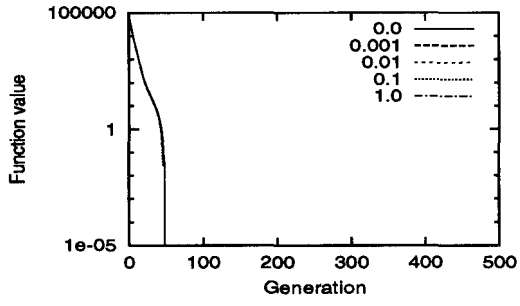



(d) cRES.

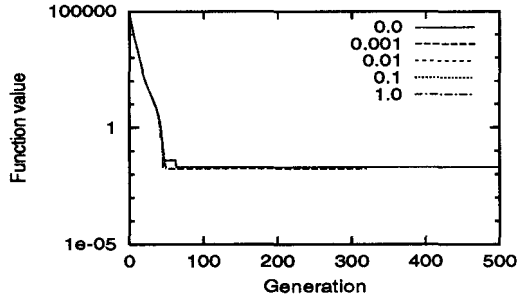
Fig. 6.39: Averaged best results on cRES for $f_5$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



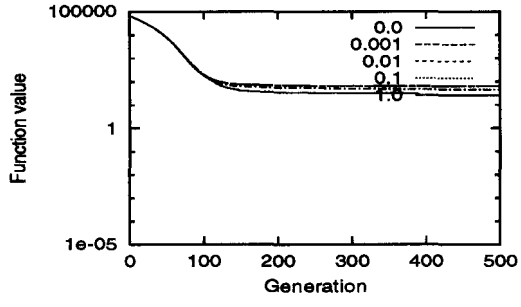

(a) II-CES.




(b) DD-CES.

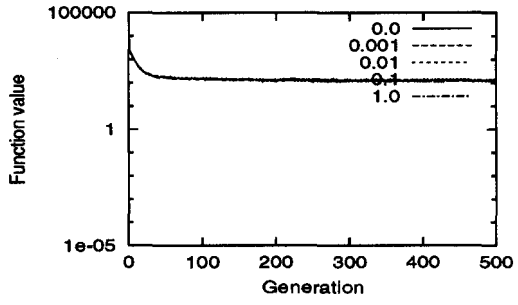



(c) D-CES.

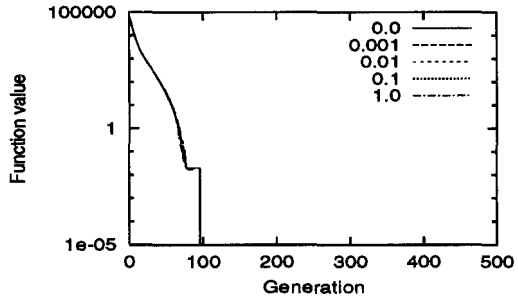



(d) CES.

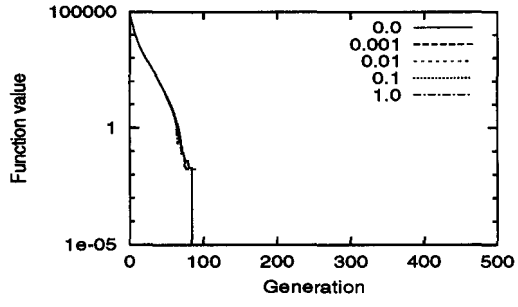
Fig. 6.40: Averaged best results on CES for $f_6$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



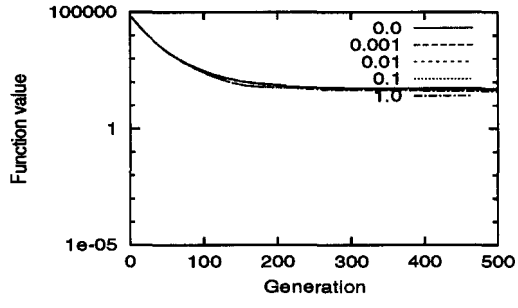

(a) II-FES.




(b) DD-FES.

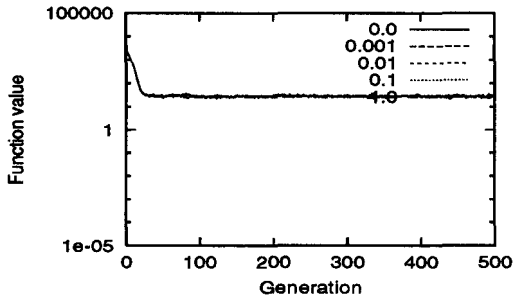



(c) D-FES.

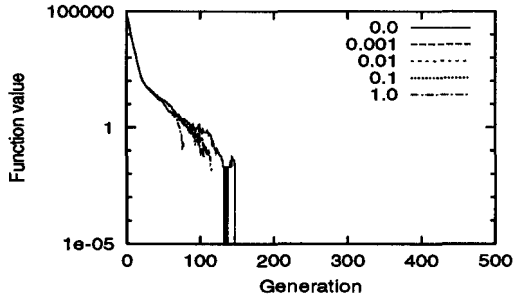



(d) FES.

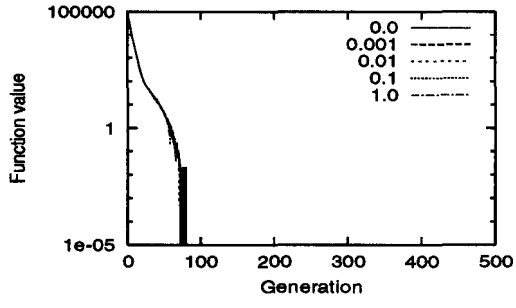
Fig. 6.41: Averaged best results on FES for $f_6$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



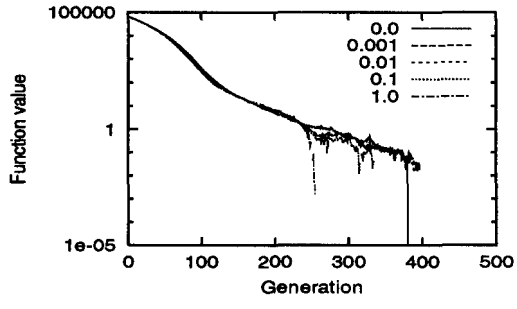

(a) II-gRES.




(b) DD-gRES.

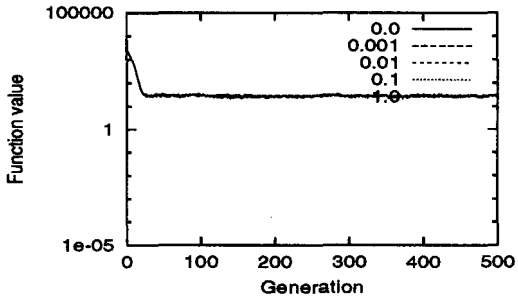



(c) D-gRES.

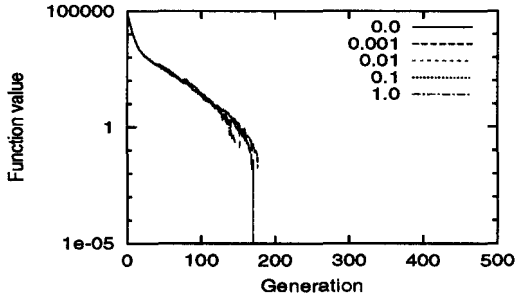



(d) gRES.

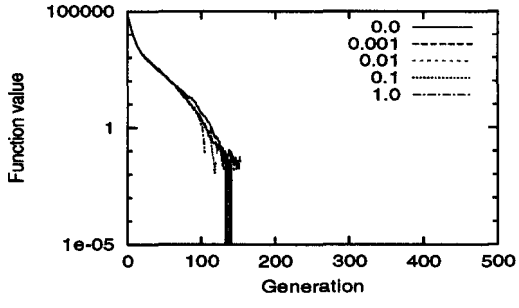
Fig. 6.42: Averaged best results on gRES for $f_6$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



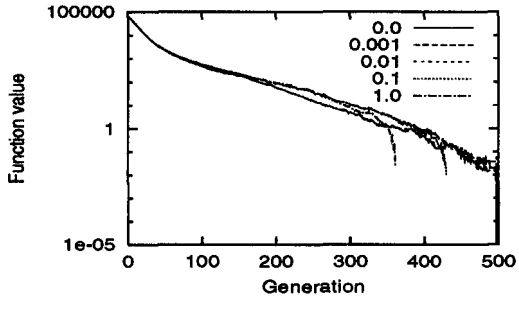

(a) II-cRES.




(b) DD-cRES.

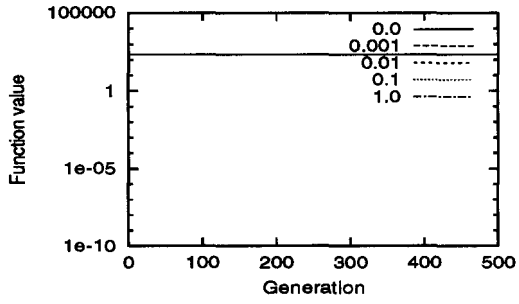



(c) D-cRES.

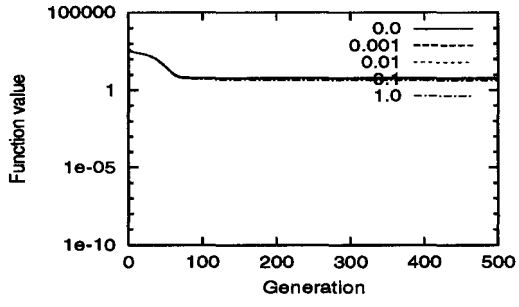



(d) cRES.

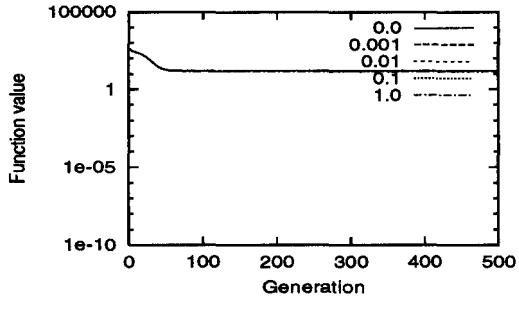
Fig. 6.43: Averaged best results on cRES for $f_6$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



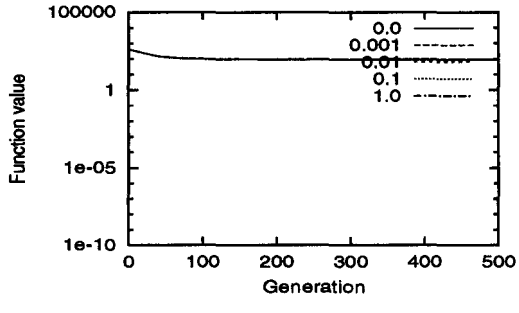

(a) II-CES.




(b) DD-CES.

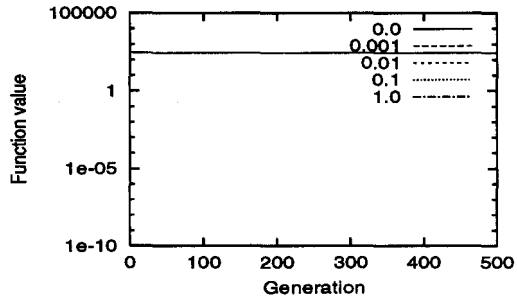



(c) D-CES.

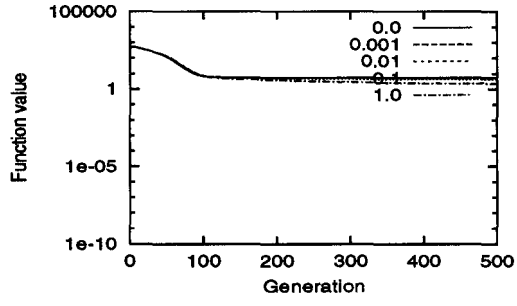



(d) CES.

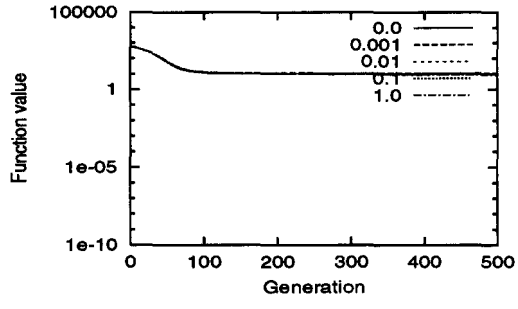
Fig. 6.44: Averaged best results on CES for $f_7$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



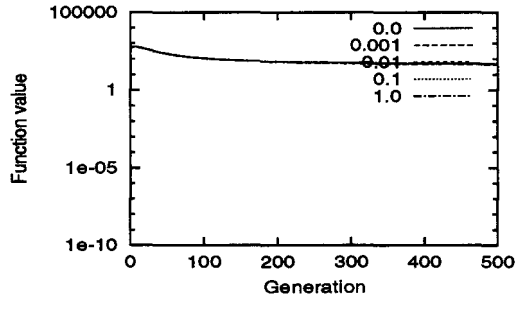

(a) II-FES.




(b) DD-FES.

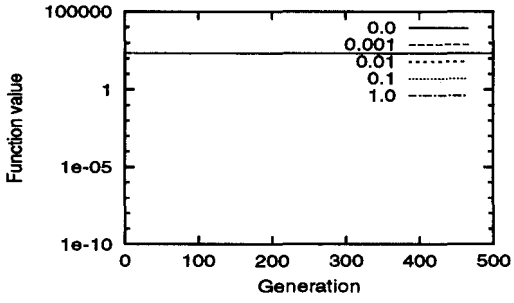



(c) D-FES.

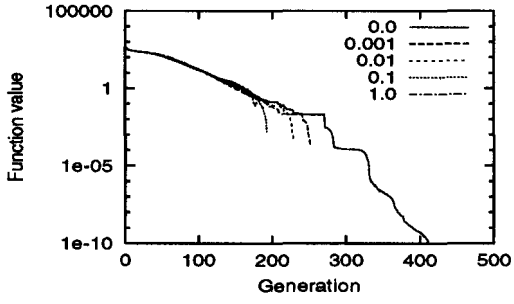



(d) FES.

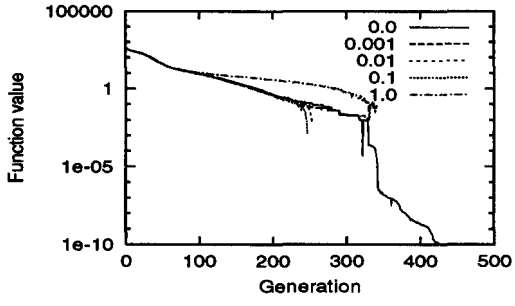
Fig. 6.45: Averaged best results on FES for $f_7$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



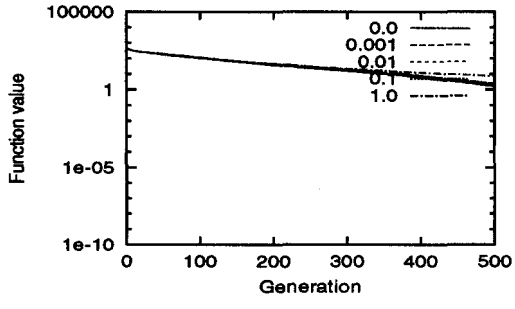

(a) II-gRES.




(b) DD-gRES.

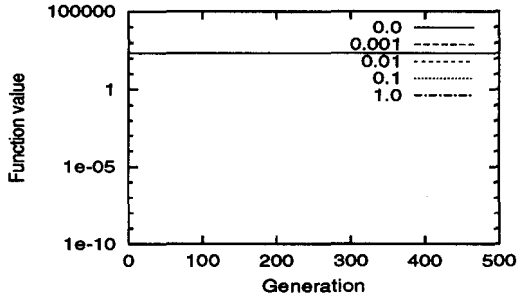



(c) D-gRES.

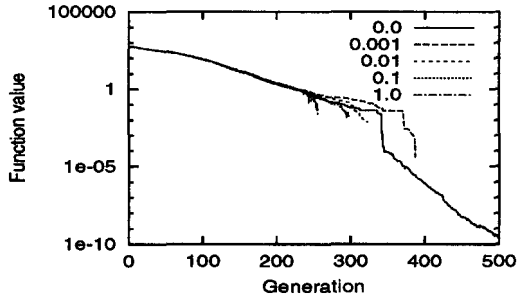



(d) gRES.

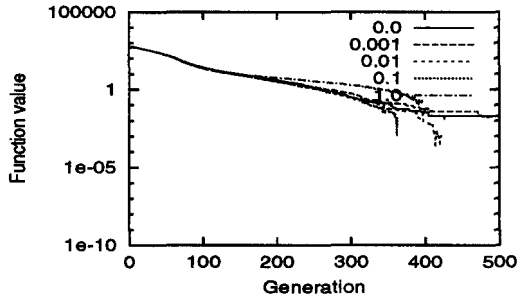
Fig. 6.46: Averaged best results on gRES for $f_7$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



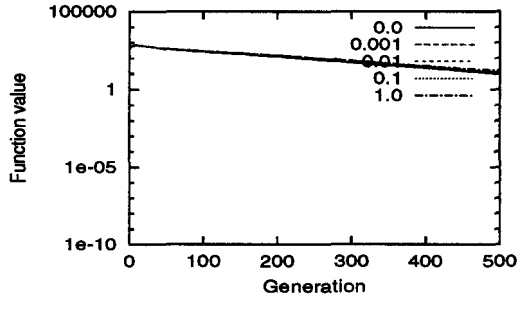

(a) II-cRES.




(b) DD-cRES.

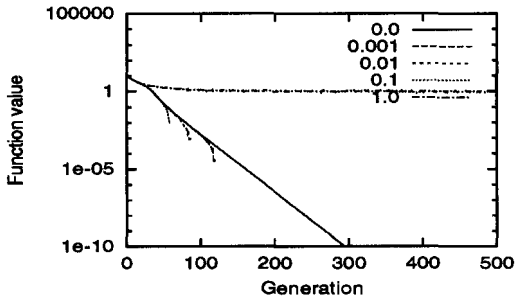



(c) D-cRES.

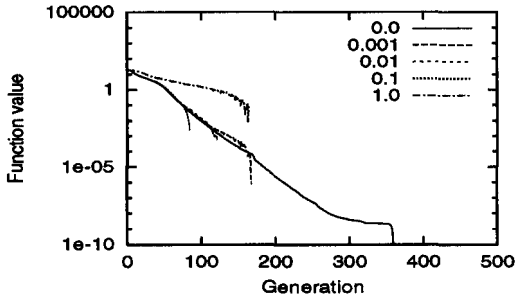



(d) cRES.

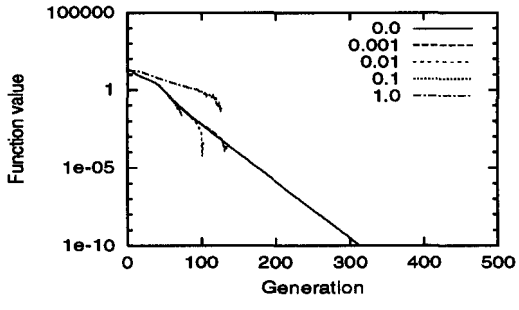
Fig. 6.47: Averaged best results on cRES for $f_7$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



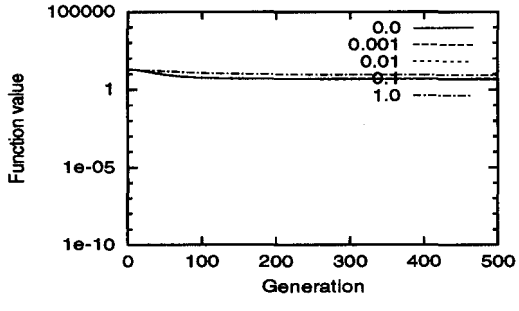

(a) II-CES.




(b) DD-CES.

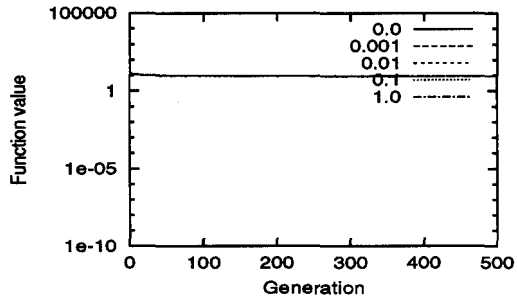



(c) D-CES.

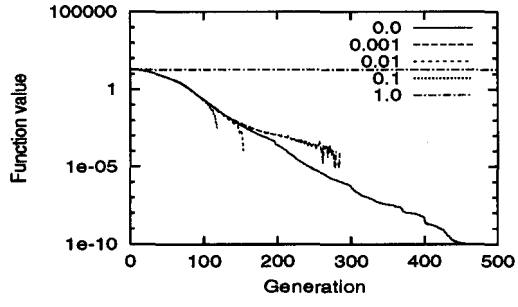



(d) CES.

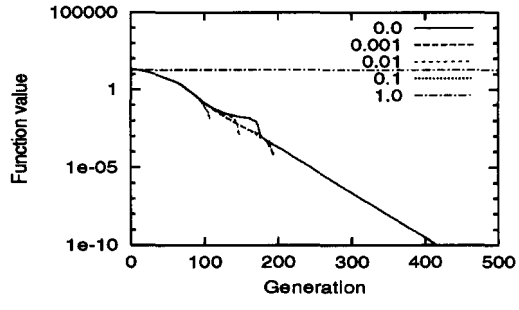
Fig. 6.48: Averaged best results on CES for $f_8$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



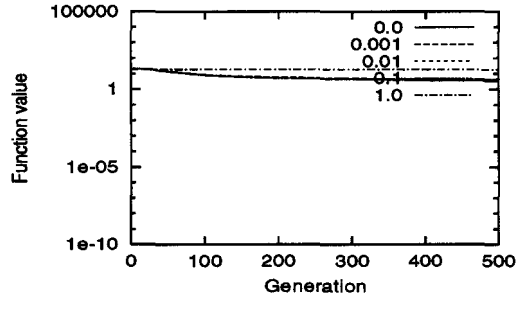

(a) II-FES.




(b) DD-FES.

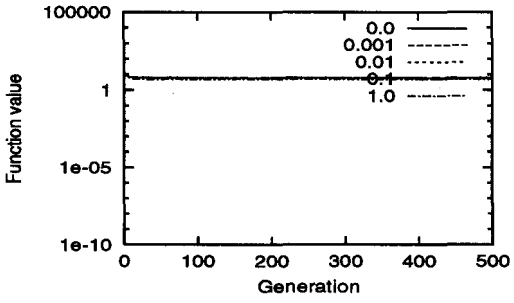



(c) D-FES.

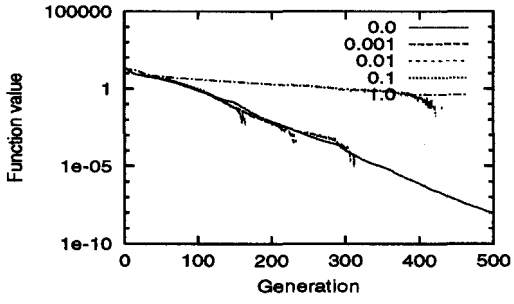



(d) FES.

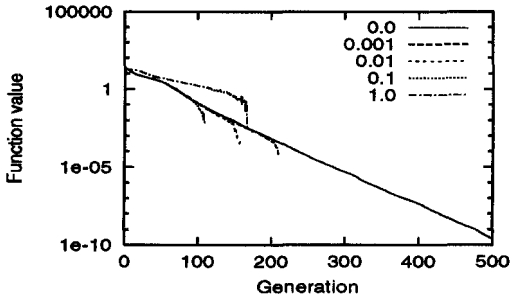
Fig. 6.49: Averaged best results on FES for $f_8$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



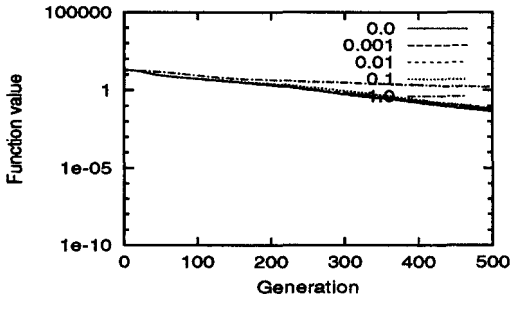

(a) II-gRES.




(b) DD-gRES.

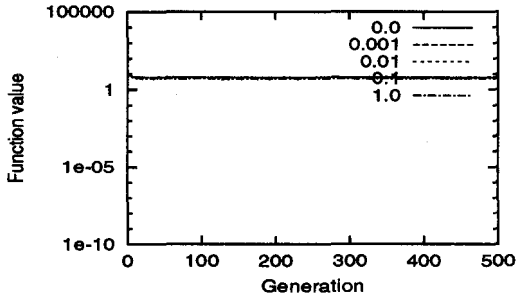



(c) D-gRES.

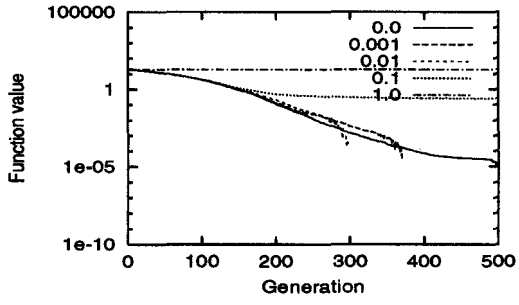



(d) gRES.

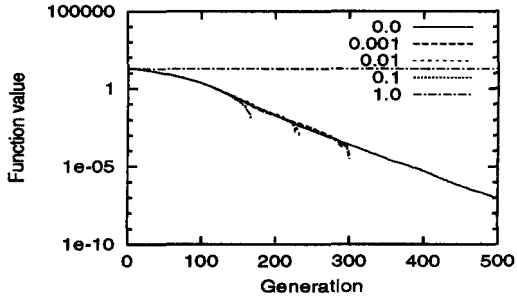
Fig. 6.50: Averaged best results on gRES for $f_8$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



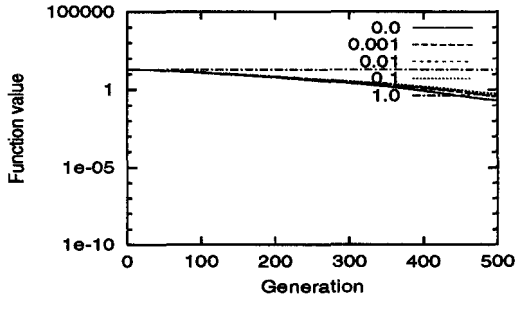

(a) II-cRES.




(b) DD-cRES.

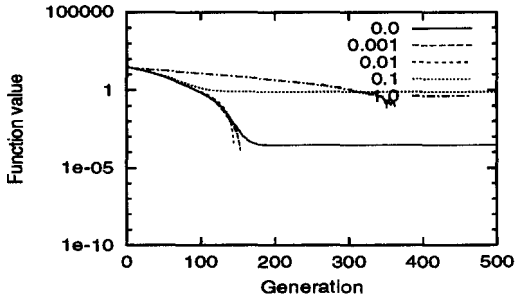



(c) D-cRES.

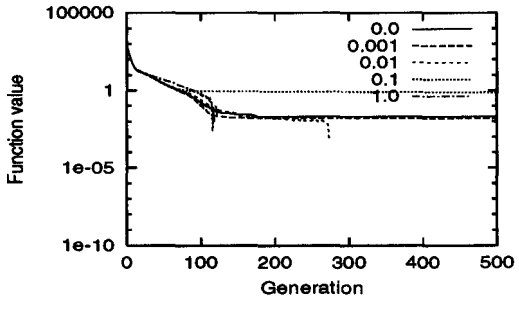



(d) cRES.

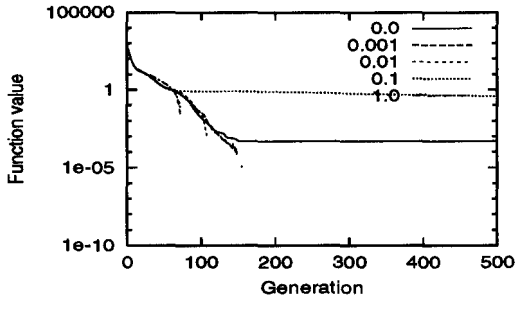
Fig. 6.51: Averaged best results on cRES for $f_8$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



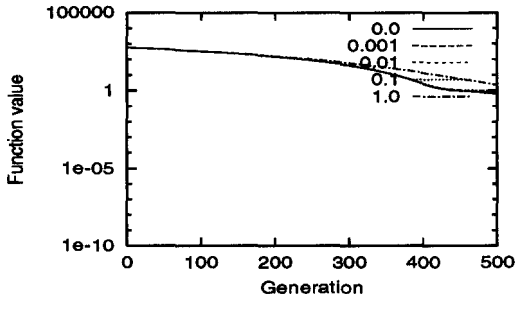

(a) II-CES.




(b) DD-CES.

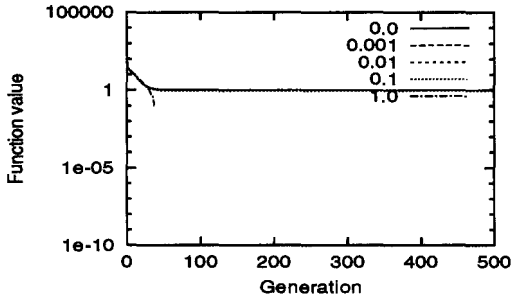



(c) D-CES.

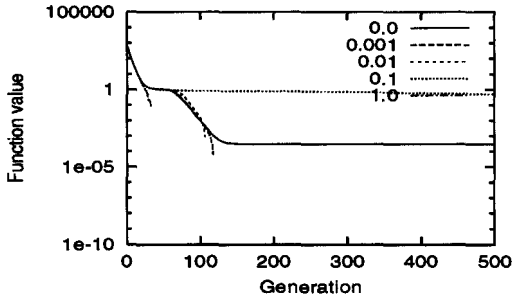



(d) CES.

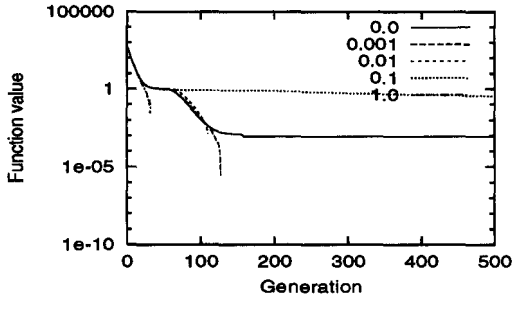
Fig. 6.52: Averaged best results on CES for $f_9$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



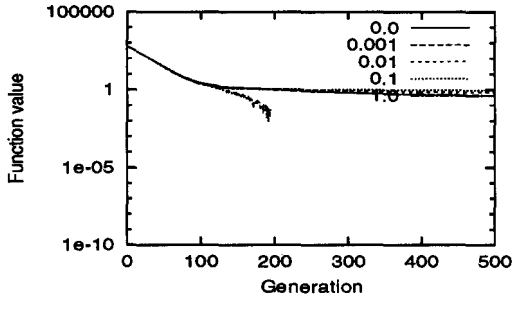

(a) II-FES.




(b) DD-FES.

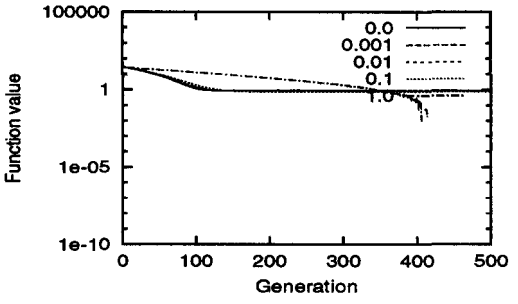



(c) D-FES.

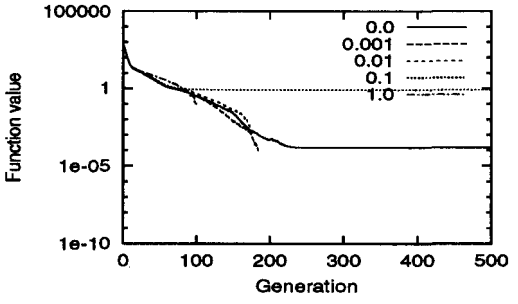



(d) FES.

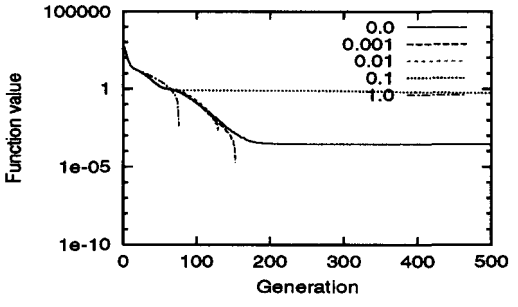
Fig. 6.53: Averaged best results on FES for $f_9$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



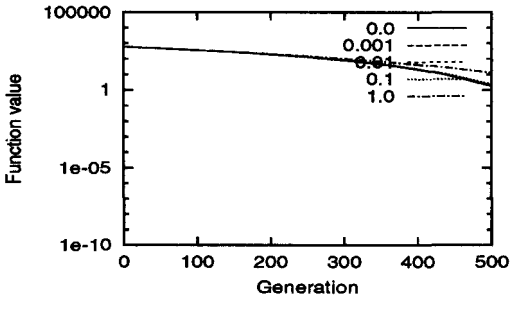

(a) II-gRES.




(b) DD-gRES.

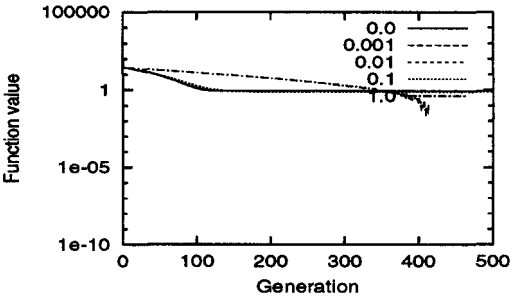



(c) D-gRES.

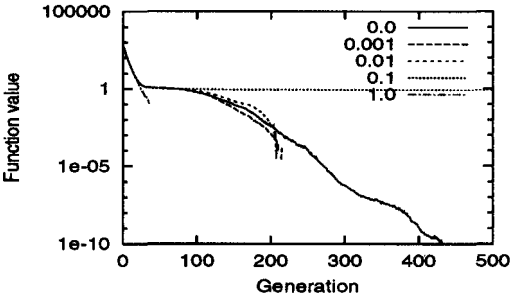



(d) gRES.

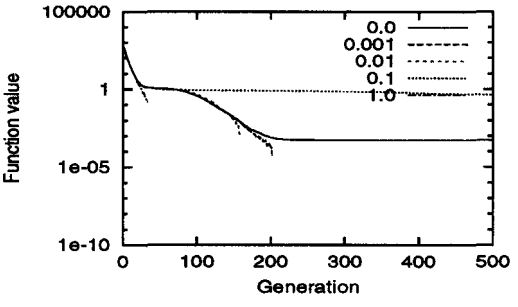
Fig. 6.54: Averaged best results on gRES for $f_9$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



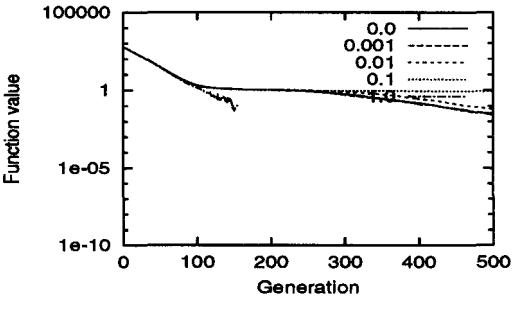

(a) II-cRES.




(b) DD-cRES.

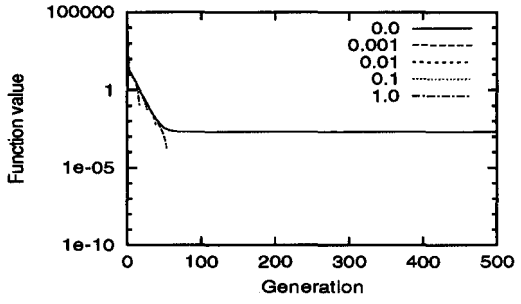



(c) D-cRES.

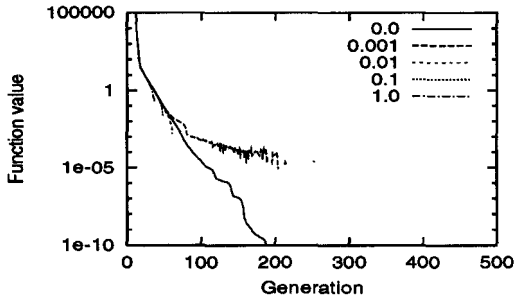



(d) cRES.

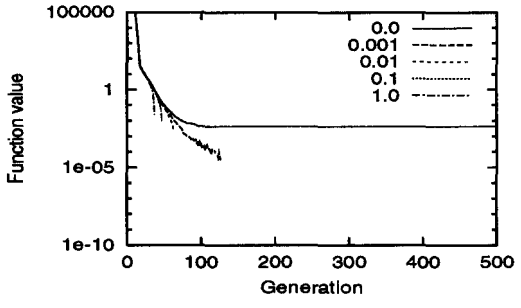
Fig. 6.55: Averaged best results on cRES for $f_9$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



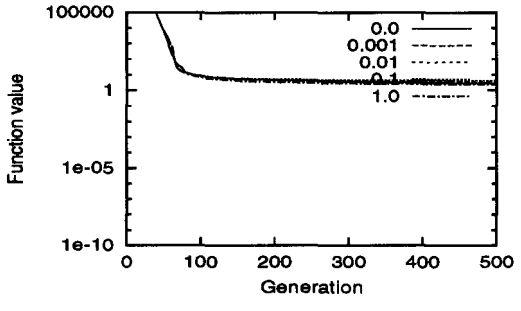

(a) II-CES.




(b) DD-CES.

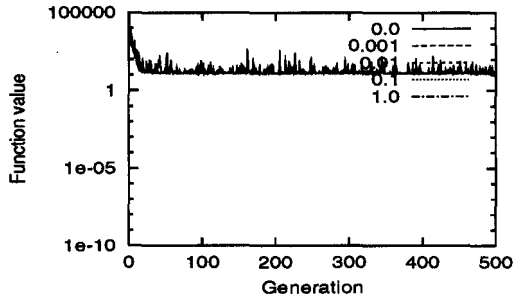



(c) D-CES.

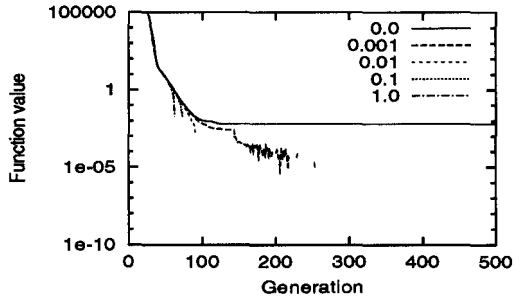



(d) CES.

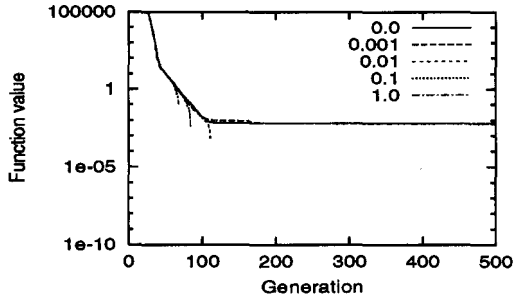
Fig. 6.56: Averaged best results on CES for $f_{10}$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



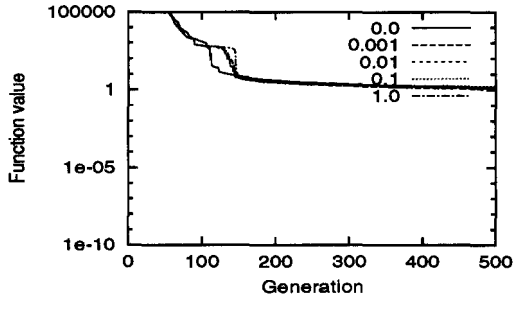

(a) II-FES.




(b) DD-FES.

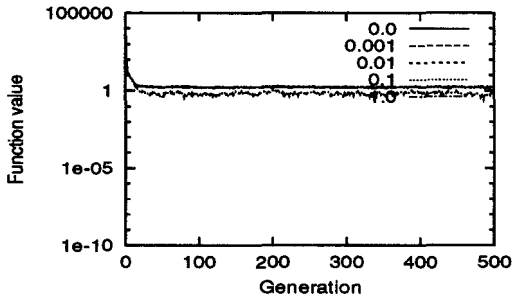



(c) D-FES.

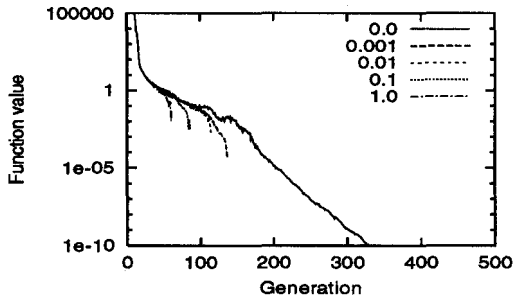



(d) FES.

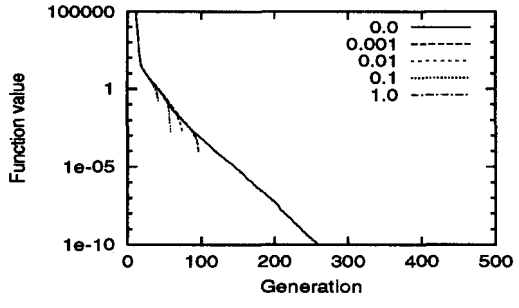
Fig. 6.57: Averaged best results on FES for $f_{10}$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



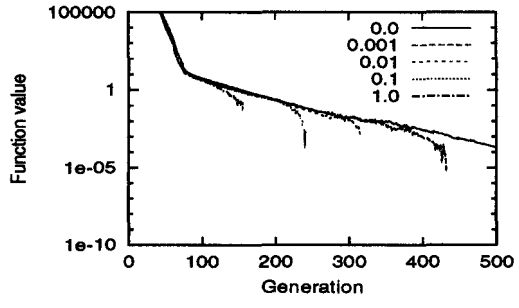

(a) II-gRES.




(b) DD-gRES.

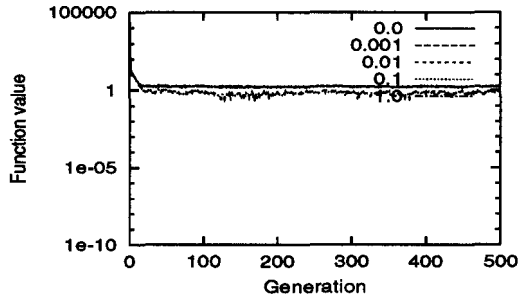



(c) D-gRES.

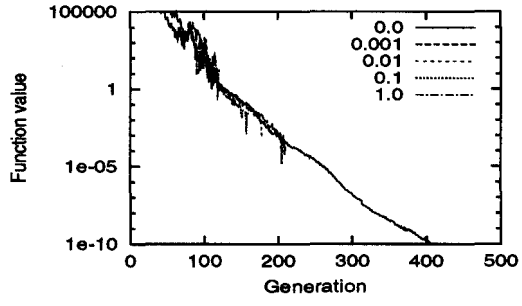



(d) gRES.

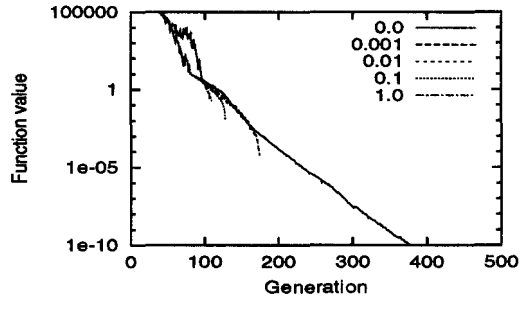
Fig. 6.58: Averaged best results on gRES for $f_{10}$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



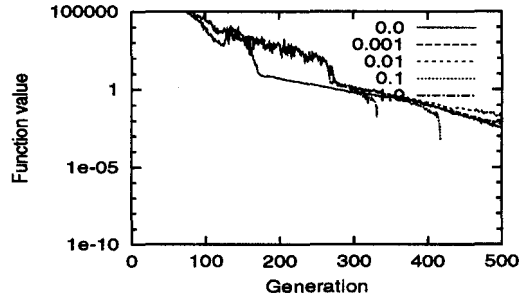

(a) II-cRES.




(b) DD-cRES.

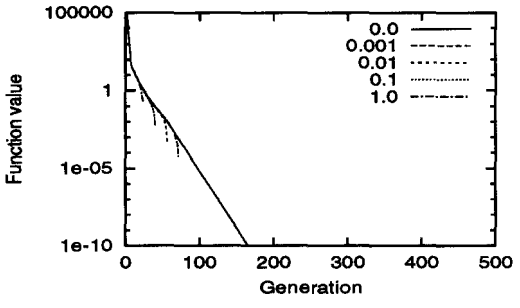



(c) D-cRES.

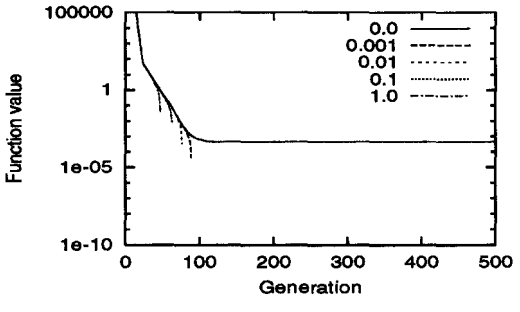



(d) cRES.

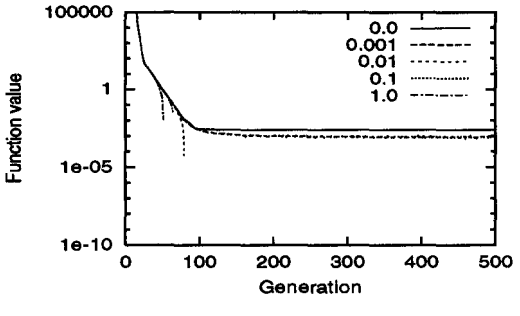
Fig. 6.59: Averaged best results on cRES for $f_{10}$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



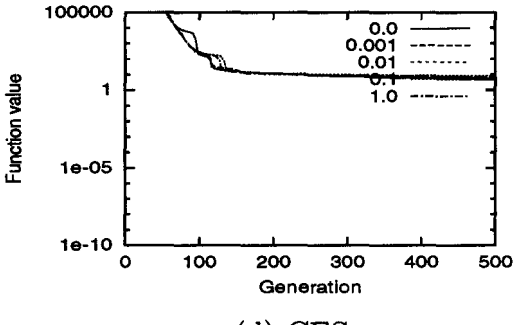

(a) II-CES.




(b) DD-CES.

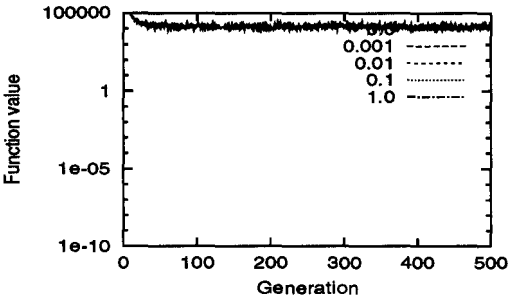



(c) D-CES.

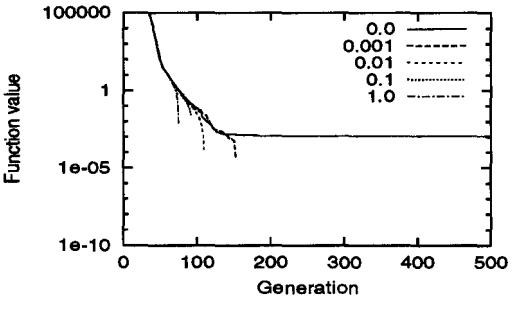



(d) CES.

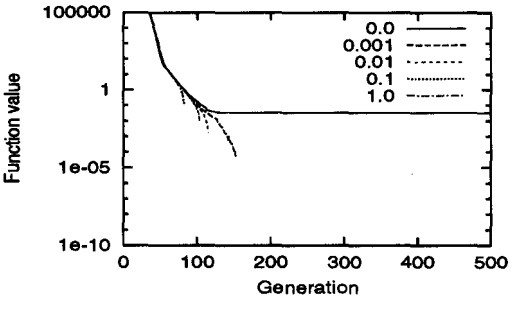
Fig. 6.60: Averaged best results on CES for $f_{11}$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



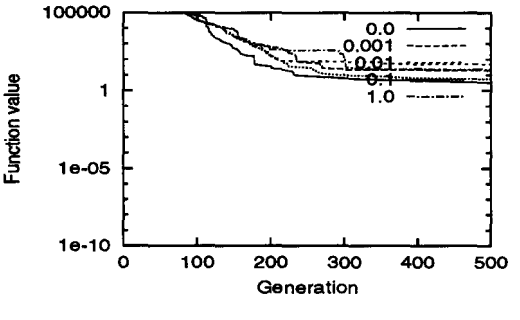

(a) II-FES.




(b) DD-FES.

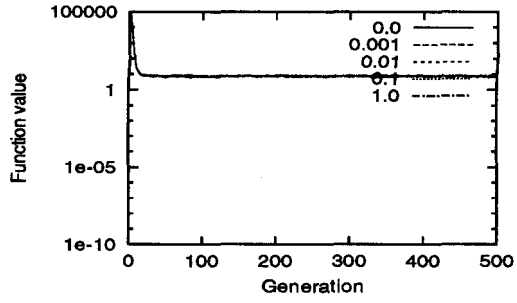



(c) D-FES.

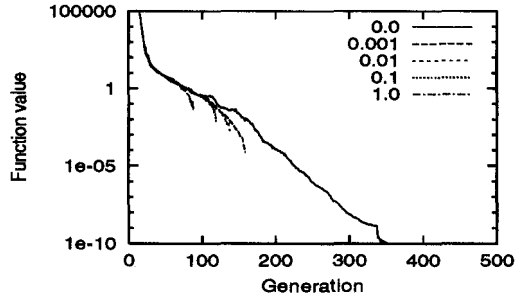



(d) FES.

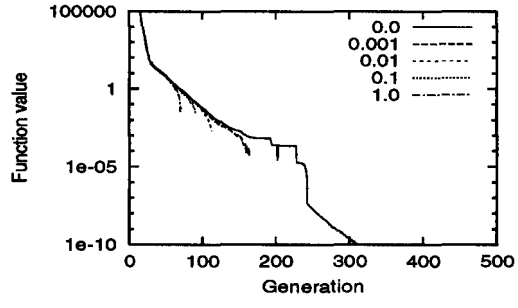
Fig. 6.61: Averaged best results on FES for $f_{11}$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



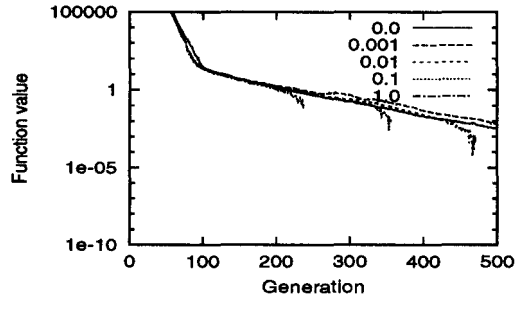

(a) II-gRES.




(b) DD-gRES.

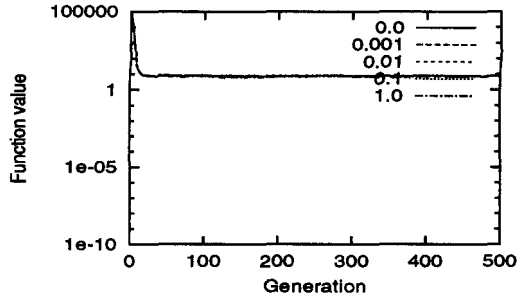



(c) D-gRES.

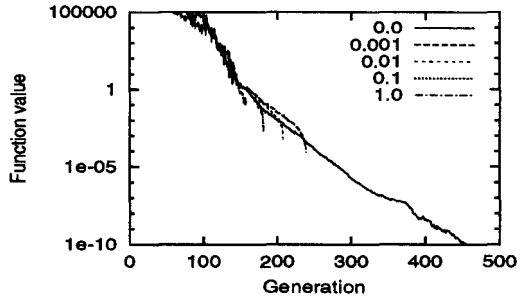



(d) gRES.

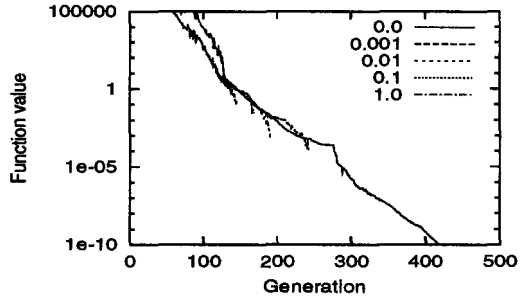
Fig. 6.62: Averaged best results on gRES for $f_{11}$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.



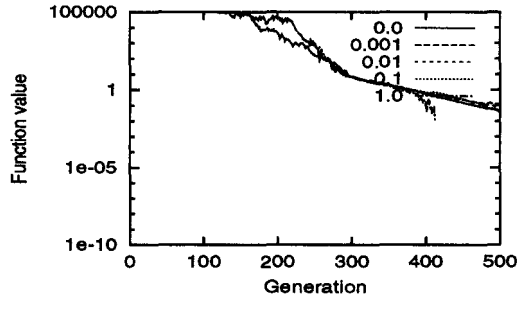

(a) II-cRES.




(b) DD-cRES.




(c) D-cRES.




(d) cRES.

Fig. 6.63: Averaged best results on cRES for $f_{11}$ with noise when the noise level($\sigma_\delta$) is 0.0, 0.001, 0.01, 0.1 and 1.0.

## 6.4 Artificial Neural Networksの構造進化への応用

工学的実問題の応用として，Evolutionary Robotics [61][50] における Evolutionary Artificial Neural Networks [115][122] を取り上げ，自律移動ロボットのナビゲーション問題を解く．2次元空間内に障害物がある環境を，自律移動ロボットがスタート位置から移動し，光源に到達することをタスクとする．その際，ロボットには1ステップ毎に10個のセンサ値が入力され，左右2個の駆動輪のモータ値が出力されるものとする．入出力の写像関係は Artificial Neural Networks で表現し，その設計を ES で行い，その際に MPR の性能を検証する．

本節で対象とするシミュレータ環境を，Fig. 5.39に示す．このシミュレータはKheperaロボットを対象に，Sussexアプローチの下にミニマルシミュレーションに基づいて構築した [62][67][96]．ロボットは，1ステップ毎に入力 $x_i(i = 0, 1, \ldots, 10 : 0 \leq x_i \leq 1.0)$ を感知し，左右2個の駆動輪のモータ出力 $y_j(j = 0, 1 : 0 \leq y_j \leq 1.0)$ を決定する．入力は Fig. 5.40 の位置にある8個の測距センサから得られる入力 $x_i(i{=}0,1,\ldots,7)$ と前方2個の光センサから得られる入力 $x_i(i{=}8,9)$ であり，Bias を入力 $x_i(i{=}10)$ とした．ロボットのスタート位置は固定されているが，配置角度はランダムに設定され，ロボットの入力 $x'_i$ ならびに最終的な出力 $y'_j$ には，Gauss分布から得られるノイズ $N(0,1)$ $(0{\leq}\ \delta \leq 1.0)$ が含まれるものとした:

$$x'_i = x_i + N(0,1) \tag{6.2}$$

$$y'_j = 18.0 \times y_j - 10.0 + 2.0 \times N(0,1) \tag{6.3}$$

ANN の重みと構造の個体表現は実数値による Indirect Encoding Scheme [122] とし，基本的な3層 Feedforward 型 ANN(Fig. 6.64) のノード数 $k(1{\leq}\ k\ {\leq}100)$ と $13k$ 個の重みを進化的に獲得させる．これは以下のように定式化される：

$$n_k = f_k(\sum_{p=0}^{10} w_{kp}x_p) \tag{6.4}$$

$$y_j = f_k(\sum_{q=0}^{k} w_{jq}n_q) \tag{6.5}$$

ただし，$n_k$ はノード $k$ の出力，$w$ は重み，$f_k$ はシグモイド関数とする．なお，$k$ の初期値は1とし変異率0.1で1個増減させる．スタートから $t$ ステップでゴールに到達した時に適応度を $800 - t$ として評価する．$(\mu, \lambda) = (30, 200)$ とし，相関突然変異を用いないこととした．なお，初期集団は $-5.0 \leq x_i(j) \leq 5.0$，$0 < \eta_i(j) \leq 3.0$ として同一のものを用い，最終世代を500(評価回数100000回) として50回独立に繰り返し計算した．戦略パラメータの最大値を $\eta_{max}$=3.0 とした．RES のオペレータの適用確率は，$P(g_{dup}) = 0.6$，$P(g_{del}) = 0.3$，$P(g_{inv}) = 0.1$ とし，冗長な戦略パラメータ数を5に設定した [83]．

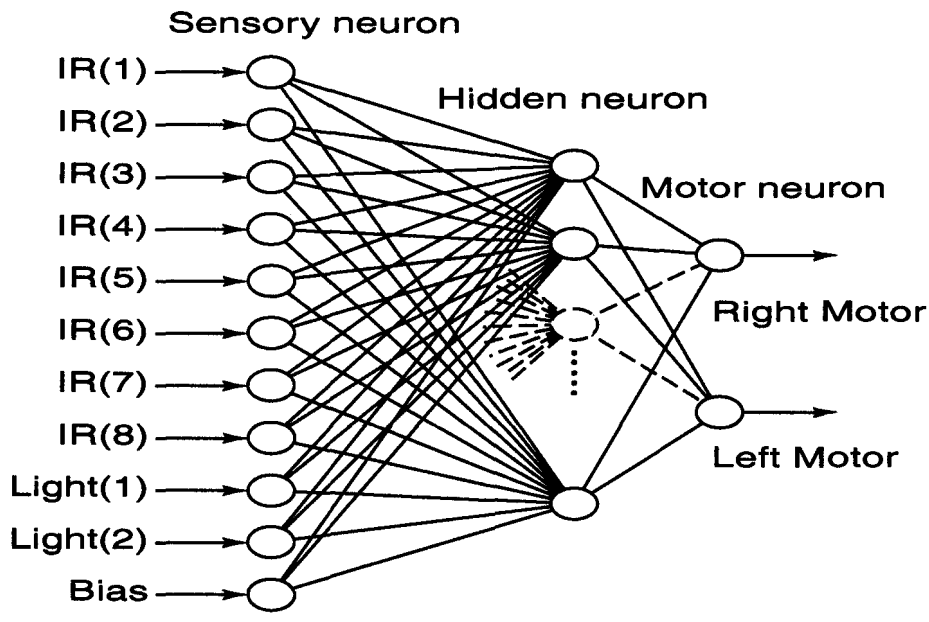
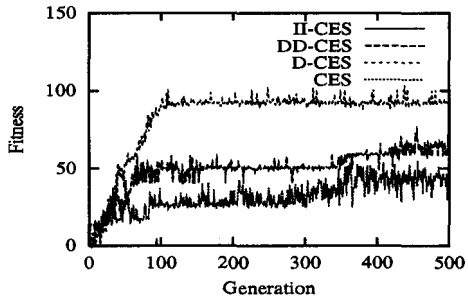


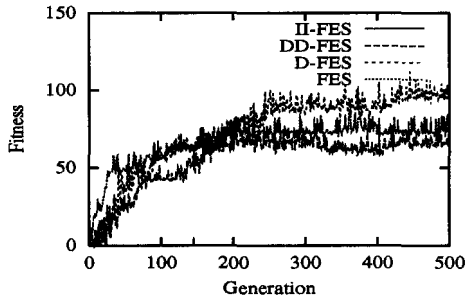

Fig. 6.64: Structure of Artificial Neural Networks.

最良個体の 50 回平均適応度を示した Fig. 6.65(a) より，CES は，D-CES，CES，DD-CES，II-CES の順に性能が向上している．FES は，Fig. 6.65(b) より，D-FES，DD-FES，FES，II-FES の順に性能が向上している．gRES は，Fig.6.65(c) より，D-gRES，DD-gRES, gRES, II-gRES の順に性能が向上している．cRES は，Fig. 6.65(d) より，D-cRES，DD-cRES，cRES，II-cRES の順に性能が向上している．

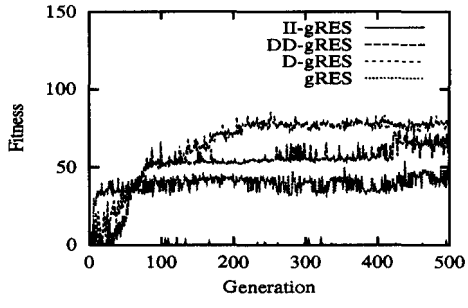
以上より，CES，FES，RES ともＤの組換え操作と相性が良いと言える．



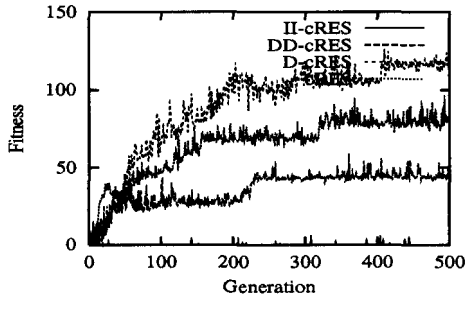

(a) CES.




(b) FES.

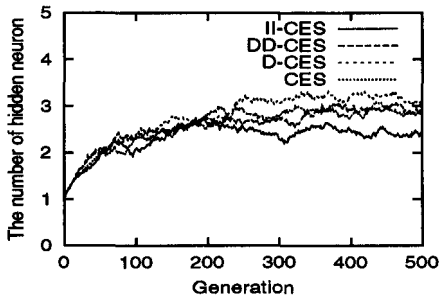



(c) gRES.

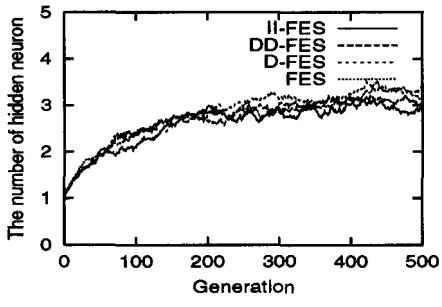



(d) cRES.

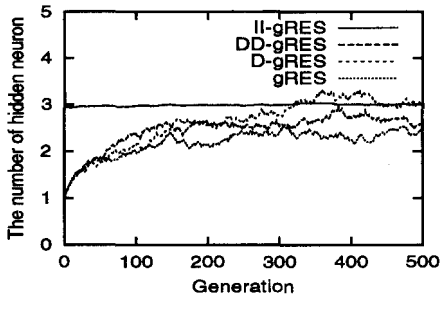
Fig. 6.65: The averaged best results for Evolutionary Artificial Neural Networks.



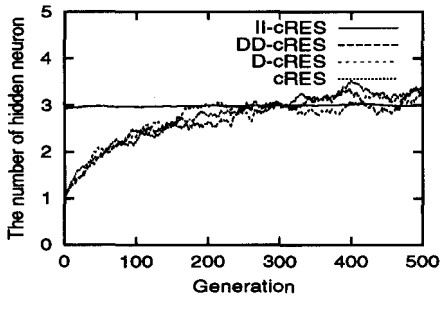

(a) CES.




(b) FES.




(c) gRES.




(d) cRES.

Fig. 6.66: The averaged number of hidden neuron for Evolutionary Artificial Neural Networks.

## 6.5 結言

本章では，進化戦略におけるMPRによる自己適応の補完を議論している．従来手法と提案手法に関して，Intermediate Recombination (IR) とDiscrete Recombination (DR) を実数値変数と戦略パラメータの両方に用いる場合のMPRの性能を検証し，その進化ダイナミクスを解析した．標準テスト関数，ノイズを含むテスト関数を対象とした計算機実験により，MPRによって進化過程における探索集団の空間的な偏りが無くなり，MPRは自己適応を補完することを明らかにした．特にMPRの中でも，実数値変数と戦略パラメータを一組とみなしてDRを用いるGlobal Combined Discrete Recombination(GCDR)は，全ての標準テスト関数，ノイズを含むテスト関数において自己適応と相性が良い．また，工学的実問題への応用として，Evolutionary Artificial Neural Networksの進化的設計を取り上げた．この設計では結合荷重だけでなく構造の設計も同時に行っているため，標準テスト関数の場合と異なり可変長の個体表現となる．そのため，適応度景観は絶えず変化し，どのような形状かは定かにならない．そのため，広く探索を行うGCDRによって性能の向上が実証されたと考えられる．

# 第7章　結論

自然進化に倣った進化型計算は，種々の最適化問題に適用できる頑健な手法として知られている．特に，進化戦略は実数値表現の目的変数と変化量を制御する戦略パラメータの2つの変数を一組として用いる個体表現法を採用していることが特徴である．これにより，進化過程で突然変異によるステップサイズの大きさが適応していくことにより，実関数最適化問題に対して優れた性能を示すと言われている．

しかしながら，高次元の非線形問題や超多峰性問題を取り扱う場合には，従来の進化戦略が持つ適応能力は十分であるとは言えない．これは戦略パラメータが大域最適解に近づく以前に非常に小さくなる現象が現れ，その結果，突然変異のステップサイズも極めて小さくなり，実質的に探索点が動かない状態に陥り集団が収束する，いわゆる早期収束が起こりやすいからである．

一般には，これを回避するために，問題の特徴に合わせた最適な下限を設定して戦略パラメータが小さくならないようにしている．しかし，この最適な下限を設定することは，(1) 本来進化型計算が目指した頑健性を失い，(2) それ以上ステップサイズが小さくならず局所探索の能力を衰えさせてしまうことから，望ましい手法であるとは言い難い．

そこで，本論文ではこの早期収束の現象を回避するため，適応能力を向上させる手法を新たに構築している．この基本的アイデアは，戦略パラメータに対し選択圧によらない確率的変化要因を加え，特定の値に固着しにくくしようとするところにある．それゆえ，各有効な戦略パラメータの他に冗長な戦略パラメータを持つ個体表現法を採用し，有効なものと冗長なものとを確率的に入れ替える新しい突然変異手法を提案している．そして，標準テスト関数に対する基本性能およびノイズに対する性能を計測し，提案手法と代表的な従来手法との比較を行うことで，提案手法の有効性を実証している．また，進化ダイナミクスの解析を行うことにより手法の特徴を明らかにし，工学的実問題への応用を通して手法の有用性を検証している．さらに，組換え操作による自己適応の補完に注目し，組換え操作と自己適応の相性を検証している．

本論文の各章は以下のように要約される．

第1章では，本論文の背景と目的，及び論文全体の構成を示している．

第2章では，本論文において対象とする進化型計算手法のうち1960年代に始まった進化戦略，進化的プログラミング，遺伝的アルゴリズムの3つの手法の概略を紹介している．次に，本論文が取り扱う実関数最適化問題を定義し，この問題領域における自己適応を説明している．特に，各手法に関する自己適応の関連研究について述べ，研

究領域を明らかにしている．そして，探索空間の次元が高くなるほど，また，適応度景観が複雑になるほど，集団が大域最適解を発見する以前に戦略パラメータが非常に小さくなる傾向が現れ，その結果，突然変異のステップサイズも極めて小さくなり，実質的に探索点が動かない状態に陥って集団が収束することが自己適応の問題点であると指摘している．

第3章では，本論文において動機付けられている生物進化メタファについて述べ，戦略パラメータの適応能力を向上させる拡張手法の基本的アイデアを説明している．提案手法は，分子進化の中立説に動機付けられており，戦略パラメータの進化に選択圧によらない確率的変化要因を加えることで，特定の値に固着しにくくしようとするものであることを説明している．そして，各有効な戦略パラメータの他に冗長な戦略パラメータを持つ個体表現法を採用し，冗長なものと有効なものを確率的に入れ替える新しい突然変異手法を定式化している．さらに，自己適応の拡張を補完するMulti-parent Recombination(MPR)について関連研究を述べ，本論文において参照する3つのMPRについて説明している．

第4章では，進化戦略の一種として考えられる進化的プログラミングに関する自己適応の拡張法について議論している．実関数最適化問題における標準テスト関数を対象とした計算機実験により，従来の進化的プログラミングは，戦略パラメータが極端に小さくなり，早期収束の状態に陥りやすいことを確認している．それに対し，第3章で定式化した提案手法は，標準テスト関数に対し，早期収束の状態を回避でき，戦略パラメータの下限設定に対する頑健性が従来手法に比べて向上している．進化ダイナミクスの観測を行った結果，従来手法に比べて提案手法は局所探索と大域探索のバランスが強化され，探索領域を伸縮しているため，遺伝的浮動の効果が発揮されていると考えられる．また，ノイズを含むテスト関数に関しても，従来手法と比較し，提案手法の有効性を実証している．さらに，工学的実問題への応用として多指ハンドよる物体の把持問題を取り上げ，Liapunovの安定条件及び接触安定条件を満たす指先位置・指先力計画の最適化を行っている．この問題は27の非線形制約条件付き最適化問題であり，超多峰性問題の一つと考えられる．特に，進化的プログラミングの特徴である最良個体が生存する点は，全ての拘束条件を一度充足すれば少なくとも1個体が拘束条件を充足したまま生存することが約束されるため，この問題に有効であると考えられる．しかしながら，従来手法を用いた計算機実験では全ての拘束条件を満たす解の発見が困難であることを確認している．それに対し，提案手法では，全ての拘束条件を満たす解を高速に発見することが可能となり，非線形制約条件付き最適化問題に対する提案手法の有用性を実証している．以上より，進化的プログラミングにおける分子進化の中立説に動機付けられた自己適応の拡張法の妥当性を示したと言える．

第5章では，進化戦略における自己適応の拡張法に関して，問題の次元に重点をおいて議論している．標準テスト関数を用いた計算機実験では，従来の進化戦略は，進化的プログラミングの場合と同様に，第2章で指摘した状態になりやすいことを明らか

にしている．これに対し，第3章で定式化した自己適応の拡張法を進化戦略に適用した手法を提案している．標準テスト関数を対象とした計算機実験により，提案手法は従来手法に比べて，戦略パラメータの下限値に対する頑健性が，向上することを実証している．提案手法の特徴を知るために突然変異率と問題の次元スケールによる進化ダイナミクスの変化を観測し，提案手法は問題の次元スケールと突然変異率のパラメータ値に対し頑健であることを確認している．また，ノイズを含むテスト関数に関してもノイズに対する提案手法の頑健性を実証している．さらに，工学的実問題への応用として，自律移動ロボットのナビゲーション問題における Continuous-Time Recurrent Neural Networks (CTRNNs) の進化的設計を取り扱っている．この CTRNNs は，実数値の入出力を取り扱うことが可能であり，連続時間を対象にしている．そのため，回路規模に対して変数は指数関数的に増大し，標準テスト関数に比べて次元の高い最適化問題となる．この高次元の問題に対しても，従来手法に対する提案手法の性能の向上を計算機実験により実証している．以上より，第4章の進化的プログラミングと比べて確定的選択とエリート保存されない点が異なる進化戦略においても，自己適応の拡張法の妥当性を示したと言える．

第6章では，進化戦略における MPR による自己適応の補完を議論している．従来手法と提案手法に関して，Intermediate Recombination (IR) と Discrete Recombination (DR) を実数値変数と戦略パラメータの両方に用いる場合の MPR の性能を検証し，その進化ダイナミクスを解析している．標準テスト関数，ノイズを含むテスト関数を対象とした計算機実験により，MPR によって進化過程における探索集団の空間的な偏りが無くなり，MPR は自己適応を補完することを明らかにしている．特に MPR の中でも，実数値変数と戦略パラメータを一組とみなして DR を用いる Global Combined Discrete Recombination(GCDR) は，全ての標準テスト関数，ノイズを含むテスト関数において自己適応と相性が良い．また，工学的実問題への応用として，Evolutionary Artificial Neural Networks の進化的設計を取り上げている．この設計では結合荷重だけでなく構造の設計も同時に行っているため，標準テスト関数の場合と異なり可変長の個体表現となる．そのため，適応度景観は絶えず変化し，どのような形状かは定かにならない．そのため，広く探索を行う GCDR によって性能の向上が実証されたと考えられる．

第7章では，本論文のまとめとして結論を述べている．

以上により，本論文は，実関数最適化問題に対して，進化型計算手法における分子進化の中立説に動機付けられた自己適応の拡張法の有効性と，MPR による自己適応の補完に関して，GCDR の有効性を実証したと結論付けられる．

# 参考文献


[1] D. V. Arnold and H.-G. Beyer (2000), "Efficiency and Mutation Strength Adaptation of the ($\mu/\mu_I$,$\lambda$)-ES in a Noisy Environment", *Proceedings of the 6th Conference on Parallel Problem Solving from Nature - PPSN VI*, Lecture Notes in Computer Science, Vol.1917, pp.39–48, Springer.

[2] D. V. Arnold and H.-G. Beyer (2001), "Investigation of the ($\mu$,$\lambda$)-ES in the Presence of Noise", *Proceedings of Congress on Evolutionary Computation (CEC'01)*, pp.332–339, IEEE Press.

[3] D. V. Arnold and H.-G. Beyer (2001), "Local Performance of the ($\mu/\mu_I$,$\lambda$)-ES in a Noisy Environment", *Foundation of Genetic Algorithms 6*, pp. 127–141, Morgan Kaufmann.

[4] D. V. Arnold and H.-G. Beyer (2002), "Local Performance of the (1+1)-ES in a Noisy Environment", *IEEE Transactions on Evolutionary Computation*, Vol.6, No.1, pp.30–41.

[5] P. J. Angeline (1995),"Adaptive and Self-Adaptive Evolutionary Computations", *Computational Intelligence: A Dynamic System Perspective*, pp.152–168, IEEE Press.

[6] P. J. Angeline (1996), " The Effects of Noise on Self-Adaptive Evolutionary Optimization", *Proceedings of the 5th Annual Conference on Evolutionary Programming*, pp.433–439, MIT Press.

[7] P. J. Angeline (1996), "Two Self-Adaptive Crossover Operations for Genetic Programming", *Advances in Genetic Programming*, Vol.2, MIT Press.

[8] T. Bäck, F. Hoffmeister and H.-P. Schwefel (1991), "A Survey of Evolution Strategies", *Proceedings of Fourth International Conference on Genetic Algorithms (ICGA'91)*, pp.2–9, Morgan Kaufmann.

[9] T. Bäck (1992), "Self-Adaptation in Genetic Algorithms", *Proceedings of the 1st European Conference on Artificial Life*, pp. 263–271, The MIT Press.

[10] T. Bäck and H.-P. Schwefel (1993), " An Overview of Evolutionary Algorithms for Parameter Optimization", *Evolutionary Computation*, Vol.1, No.1, pp.1–23.

[11] T. Bäck, G. Rudolph and H.-P. Schwefel (1993), "Evolutionary Programming and Evolution Strategies: Similarities and Differences", *Proceedings of the 2nd Annual Conference on Evolutionary Programming*, pp. 11-22.

[12] T. Bäck and U. Hammel (1994), "Evolution Strategies Applied to Perturbed Objective Functions", *Proceedings of the First IEEE Conference on Evolutionary Computation*, pp.40–45, IEEE Press.

[13] T. Bäck (1996), *Evolutionary Algorithms in Theory and Practice*, Oxford University Press.

[14] T. Bäck, U. Hammel and H.-P. Schwefel (1997), "Evolutionary Computation: Comments on the History and Current State", *IEEE Transactions on Evolutionary Computation*, Vol.1, No.1, pp.3–17.

[15] T. Bäck and A. E. Eiben (1999), "Generalizations of Intermediate Recombination in Evolution Strategies", *Proceedings of Congress on Evolutionary Computation (CEC'99)*, pp.1566–1573, IEEE Press.

[16] M. A. Bedau, E. Snyder C. T. Brown and N. H. Packard (1997), "A Comparison of Evolutionary Activity in Artificial Evolving Systems and in the Biosphere", *Proceedings of the Fourth European Conference on Artificial Life*, MIT Press.

[17] M. A. Bedau, E. Snyder and N. H. Packard (1998), "A Classification of Long-term Evolutionary Dynamics", *Proceedings of Artificial life 6*, pp.228–237, MIT Press.

[18] R. D. Beer, and J.C. Gallagher (1992), "Evolving Dynamical Neural Networks for Adaptive Behavior", *Adaptive Behavior* Vol.1, No.1, pp.91–122.

[19] R. D. Beer (1995), " On the Dynamics of Small Continuous-Time Recurrent Neural Networks", *Adaptive Behavior*, Vol.3, No.4, pp.471–511.

[20] R. D. Beer (1996), "Toward the Evolution of Dynamical Neural Networks for Minimally Cognitive Behavior", *From animals to animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, pp.421–429, MIT Press.

[21] H.-G. Beyer (1993), "Toward a Theory of Evolution Strategies: Some Asymptotical Results from the $(1,^{+}\lambda)$- Theory ", *Evolutionary Computation*, Vol.1, No.2, pp.165–188.

[22] H.-G. Beyer (1994), "Toward a Theory of Evolution Strategies: The $(\mu,\lambda)$-Theory ", *Evolutionary Computation*, Vol.2, No.4, pp.381–407.

[23] H.-G. Beyer (1995), "Toward a Theory of Evolution Strategies: On the Benefits of $(\mu/\mu,\lambda)$-Theory ", *Evolutionary Computation*, Vol.3, No.1, pp.81–111.

[24] H.-G. Beyer (1996), "Toward a Theory of Evolution Strategies: Self-adaptation", *Evolutionary Computation*, Vol.3, No.3, pp.311–347.

[25] H.-G. Beyer (1998), "Mutate Large, But Inherit Small! On the Analysis of Rescaled Mutations in $(\tilde{1},\tilde{\lambda})$-ES with Noisy Fitness Data", *Proceedings of 5th Conference on Parallel Problem Solving from Nature - PPSN V*, Lecture Notes in Computer Science, Vol.1498, pp.109–118, Springer.

[26] H.-G. Beyer and K. Deb (1999), "On the Analysis of Self-Adaptive Evolutionary Algorithms", *Technical Report No. CI-69/99, Department of Computer Science, University of Dortmund, Germany.*

[27] H.-G. Beyer and D. V. Arnold (1999), "Fitness Noise and Localization Errors of the Optimum in General Quadratic Fitness Models", *Proceedings of Genetic and Evolutionary Computation Conference (GECCO'99)*, pp.817–824, Morgan Kaufmann.

[28] H.-G. Beyer and K. Deb (2000), "On the Desired Behaviors of Self-Adaptive Evolutionary Algorithms", *Proceedings of 6th Conference on Parallel Problem Solving from Nature - PPSN VI*, Lecture Notes in Computer Science, Vol.1917, pp. 59–68, Springer.

[29] H.-G. Beyer (2001), *The Theory of Evolution Strategies*, Springer.

[30] H.-G. Beyer and K. Deb (2001), "On Self-Adaptive Features in Real-Parameter Evolutionary Algorithms", *IEEE Transactions on Evolutionary Computation*, Vol.5, No.3, pp.250–270.

[31] H.-G. Beyer (2001),"On the Performance of $(1,\lambda)$-Evolution Strategies for the Ridge Function Class", *IEEE Transactions on Evolutionary Computation*, Vol.5, No.3, pp.218–235.

[32] M. Capcarrere, M. Tomassini, A. Tettamanzi and M. Sipper (1999), "A Statistical Study of a Class of Cellular Evolutionary Algorithms", *Evolutionary Computation*, Vol.7, No.3, pp.255–274.

[33] U. Chakraborty, K. Deb and M. Chakraborty (1996), "Analysis of Selection Algorithms: A Markov Chain Approach", *Evolutionary Computation*, Vol.4, No.2, pp.133–167.

[34] M. Chang, K. Ohkura and K. Ueda (2001), "Some Experimental Observation of $(\mu/\mu,\lambda)$-Evolution Strategies", *Proceedings of Congress on Evolutionary Computation (CEC'01)*, pp.663–670, IEEE Press.

[35] K. Chellapilla and D. B. Fogel (1997), " Exploring Self-Adaptive Methods to Improve the Efficiency of Generating Approximate Solutions to Traveling Salesman Problems Using Evolutionary Programming, " *Proceedings of the 6th Annual Conference on Evolutionary Programming*, Lecture Notes in Computer Science, Vol.1213, pp.13–16, Springer.

[36] K. Chellapilla (1998), "Combining Mutation Operators in Evolutionary Programming", *IEEE Transactions on Evolutionary Computation*, Vol.2, No.3, pp.91–96.

[37] K. Chellapilla and D. B. Fogel (1998), " Two New Mutation Operators for Enhanced Search and Optimization in Evolutionary Programming " *International Symposium on Optical Science, Engineering, and Instrumentation, Conference 3165: Applications of Soft Computing*, SPIE.

[38] K. Chellapilla and D. B. Fogel (1999), "Fitness Distribution in Evolutionary Computation: Analysis of Local Extrema in the Continuous Domain", *Proceedings of Congress on Evolutionary Computation (CEC'99)*, pp.1885–1892.

[39] K. Chellapilla and D. B. Fogel (1999), "Fitness Distributions in Evolutionary Computation: Motivation and examples in the Continuous Domain", *Bio Systems*, Vol.54, pp.15–29, Elsevier.

[40] J. C. Culberson (1998), "On the Futility of Blind Search: An Algorithmic View of N̈o Free Luncḧ", *Evolutionary Computation*, Vol.6, No.2, pp.109–127.

[41] K. Deb and H.-G. Beyer (1999), "Self-adaptation in Real-Parameter Genetic Algorithms with Simulated Binary Crossover", *Proceedings of Genetic and Evolutionary Computation Conference (GECCO'99)*, pp.172–179, Morgan Kaufmann.

[42] K. Deb and H.-G. Beyer (2001), "Self-Adaptive Genetic Algorithms with Simulated Binary Crossover", *Evolutionary Computation*, Vol.9, No.2, pp.197–221.

[43] A. E. Eiben, P-E. Raué and Zs. Ruttkay (1994), "Genetic Algorithms with Multiparent Recombination", *Proceedings of the 3rd Conference on Parallel Problem Solving from Nature - PPSN III*, Lecture Notes in Computer Science, Vol.866, pp.78–87, Springer.

[44] A. E. Eiben and C. A. Schippers (1996), "Multi-parent's Niche: N-ary Crossovers on NK-landscapes", *Proceedings of the 4th Conference on Parallel Problem Solving from Nature - PPSN IV*, Lecture Notes in Computer Science, Vol.1141, pp.319–328, Springer.

[45] A. E. Eiben (1997), "Multi-parent Recombination" *Handbook of Evolutionary Computation*, C3.3.7, Oxford University Press.

[46] A. E. Eiben and T. Bäck (1998), "An Empirical Investigation of Multiparent Recombination Operators in Evolution Strategies", *Evolutionary Computation*, Vol.5, No.3, pp.347–365.

[47] A. E. Eiben and A. Schippers (1998), " On Evolutionary Exploration and Exploitation", *Fundamenta Informaticae*, Vol.35 (1-4), pp.35–50, IOS press.

[48] A. E. Eiben, R. Hinterding and Z. Michalewicz (1999), "Parameter Control in Evolutionary Algorithms", *IEEE Transactions on Evolutionary Computation*, Vol.3, No.2, pp.124–141.

[49] L. J. Eshelman and J. D. Schaffer (1993), "Real-Coded Genetic Algorithms and Interval-Schemata", *Foundation of Genetic Algorithms 2*, pp.187–202, Morgan Kaufmann.

[50] D. Floreano and F. Mondada (1994), "Automatic Creation of an Autonomous Agent: Genetic Evolution of a Neural-network Driven Robot", *From Animals to Animats 3: Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, pp.421–430, MIT Press.

[51] D. B. Fogel (1995), *Evolutionary Computation Toward a New Philosophy of Machine Intelligence*, IEEE Press.

[52] D. B. Fogel and H.-G. Beyer (1996), " A Note on the Empirical Evaluation of Intermediate Recombination", *Evolutionary Computation*, Vol.3, No.4, pp.491–495.

[53] D. B. Fogel, E. C. Wasson, E. M. Boughton, V. W. Porto and P. J. Angeline PJ (1998), "Linear and Neural Models for Classifying Breast Cancer", *IEEE Transactions on Medical Imaging*, Vol.17, No.3, pp.485–488.

[54] L. J. Fogel, P. J. Angeline and D. B. Fogel(1995), "An Evolutionary Programming Approach to Self-Adaptation in Finite State Machine", *Proceedings of the 4th Annual Conference on Evolutionary Programming*, pp.355–365, MIT Press

[55] D. K. Gehlhaar and D. B. Fogel (1996), "Tuning Evolutionary Programming for Conformationally Flexible Molecular Docking", *Proceedings of the 5th Annual Conference on Evolutionary Programming*, pp 419–429, MIT Press.

[56] D. Goldberg (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley.

[57] L. Gruenz and H.-G. Beyer (1999), "Some Observations on the Interaction of Recombination and Self-Adaptation in Evolution Strategies", *Proceedings of Congress on Evolutionary Computation (CEC'99)*, pp.639–645, IEEE Press.

[58] U. Hammel and T. Bäck (1994), "Evolution Strategies on Noisy Functions: How to Improve Convergence Properties", *Proceedings of Parallel Problem solving from Nature III - (PPSN III)*, Lecture Notes in Computer Science, Vol.866, pp.418–427, Springer.

[59] N. Hansen (2000), "Invariance, Self-Adaptation and Correlated Mutations in Evolution Strategies", *Proceedings of 6th Conference on Parallel Problem Solving from Nature - PPSN VI*, Lecture Notes in Computer Science, Vol.1917, pp.355–364, Springer.

[60] W. E. Hart (1996), "A Theoretical Comparison of Evolutionary Algorithms and Simulated Annealing", *Proceedings of the 5th Annual Conference on Evolutionary Programming*, pp.147–154, MIT Press.

[61] I. Harvey, P. Husbands and D. Clif (1992), "Issues in Evolutionary Robotics", *From Animals to Animats 2: Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, MIT Press.

[62] I. Harvey, P. Husbands, D. Clif, A. Thompson, and N. Jakobi (1997), "Evolutionary Robotics: the Sussex Approach", *Robotics and autonomous systems*, Vol.20, pp.205–224.

[63] R. Hinterding (1995), "Gaussian Mutation and Self-adaption for Numeric Genetic Algorithms", *Proceedings of the 2nd International Conference on Evolutionary Computation*, pp. 384–389, IEEE Press.

[64] R. Hinterding (1997), "Self-adaptation using Multi-chromosomes", *Proceedings of the 4th IEEE International Conference on Evolutionary Computation*, pp.87–91, IEEE Press.

[65] R. Hinterding, Z. Michalewicz and A. E. Eiben (1997), "Adaptation in Evolutionary Computation: A Survey", *Proceedings of the 4th IEEE International Conference on Evolutionary Computation*, pp.65–69, IEEE Press.

[66] J. H. Holland (1992), *Adaptation in Natural and Artificial Systems*, MIT Press.

[67] N. Jacobi, P. Husbands and I. Harvey (1995), "Noise and the Reality Gap: The Use of Simulation in Evolutionary Robotics", *Third European Conference on Artificial Life(ECAL95)*, pp.704–720, Springer.

[68] C. Kappler (1996), "Are Evolutionary Algorithms Improved by Large Mutations?", *Proceedings Parallel Problem Solving from Nature IV - (PPSN IV)*, (H.-M. Voigt, W. Ebeling, I. Rechenberg and H.-P. Schwefel, eds.), Lecture Notes in Computer Science, Vol.1141, pp.346–355, Springer.

[69] J.-H. Kim and H. Myung (1997), "Evolutionary Programming Techniques for Constrained Optimization Problems", *IEEE Transactions on Evolutionary Computation*, Vol.1, No.2, pp.129–140.

[70] M. Kimura (1983), *The Neutral Theory of Molecular Evolution*, Cambridge University Press.

[71] F. Kursawe (1995), "Toward Self-adapting Evolution Strategies", *Proceedings of 2nd IEEE Conference Evolutionary Computation*, pp.283–288, IEEE Press.

[72] C.-Y. Lee and Y. Song (1999), "Evolutionary Programming using the Levy Probability Distribution", *Proceedings of Genetic and Evolutionary Computation Conference (GECCO'99)*, pp.886–893, Morgan Kaufmann.

[73] K.-H. Liang, X. Yao, Y. Liu, C. Newton and D. Hoffman (1998), "An Experimental Investigation of Self-adaptation in Evolutionary Programming", *Proceedings of the 7th Annual Conference on Evolutionary Programming*, pp.291–300, Springer.

[74] K.-H. Liang, X. Yao and C. Newton (1998), "Dynamic Control of Adaptive Parameters in Evolutionary Programming", *Proceedings of the 2nd Asia-Pacific Conference on Simulated Evolution and Learning(SEAL'98)*, Lecture Note in Computer Science, Vol.1585, pp.42–49, Springer.

[75] K.-H. Liang, X. Yao and C. S. Newton (2001), "Adapting Self-adaptive Parameters in Evolutionary Algorithms", *Applied Intelligence*, Vol.15, No.3, pp.171–180, Kluwer Academic Publishers.

[76] Y. Matsumura, K. Ohkura and K. Ueda (1999), "Evolutionary Programming with Non-coding Segments for Real-valued Function Optimization", *Proceedings of 1999 IEEE International Conference on Systems, Man and Cybernetics (SMC'99)*, Vol.4, pp.242–247.

[77] Y. Matsumura, K. Ohkura and K. Ueda (2000), "Statistical Characteristics of Evolution Strategies", *Proceedings of 6th Conference on Parallel Problem Solving from Nature - PPSN VI*, Lecture Notes in Computer Science, Vol.1917, pp.119–128, Springer.

[78] Y. Matsumura, K. Ohkura and K. Ueda (2001), "Evolutionary Dynamics of Evolutionary Programming in Noisy Environment", *Proceedings of Congress on Evolutionary Computation (CEC2001)*, pp.17–24, IEEE Press.

[79] Y. Matsumura, K. Ohkura and K. Ueda (2001), "The Effect of Multi-parent Recombination on Evolution Strategies for Noisy Objective Functions", *Proceedings of 2001 International Conference on Computational Intelligence for Modeling, Control and Automation (CIMCA2001)*, pp.227–236.

[80] Y. Matsumura, K. Ohkura and K. Ueda (2001), "The Effect of Multi-parent Recombination on Evolution Strategies", *Proceedings of The Fourth IFAC Symposium on Intelligent Autonomous Vehicle (IAV2001)*, pp.127–132.

[81] Y. Matsumura, K. Ohkura and K. Ueda (2001), "Multi-recombinant Evolution Strategies for Real-Valued Function Optimization", *Proceedings of ANNIE '01 Smart Engineering Systems Design*, pp159–164.

[82] Y. Matsumura, K. Ohkura and K. Ueda (2002), "$(\mu/\mu,\lambda)$-Evolution Strategies for Noisy Objective Functions", *Proceedings of International Workshop on Emergent Synthesis (IWES)*, pp.13–23.

[83] Y. Matsumura, K. Ohkura and K. Ueda (2002), "Advantages of Global Discrete Recombination in ($\mu/\mu$,$\lambda$)-Evolution Strategies", *Proceedings of IEEE World Congress on Computational Intelligence (WCCI)*, pp.1848–1853.

[84] Z. Michalewicz and D. B. Fogel (2000), "How to Solve It: Modern Heuristics", Springer.

[85] H. Mühlenbein and H.-M. Voigt (1995), "Gene Pool Recombination in Genetic Algorithms", *Proceedings of the Metaheuristics Conference*, Kluwer Academic Publishers.

[86] V. Nissen and J. Propach (1998), "Optimization with Noisy Function", *Proceedings of 5th Conference on Parallel Problem Solving from Nature - PPSN V*, Lecture Notes in Computer Science, Vol.1498, pp.159–168, Springer.

[87] M. Nei (1987), *Molecular Evolutionary Genetics*, Columbia University Press.

[88] K. Ohkura and K. Ueda (1997), "An Extended Framework for Overcoming Premature Convergence", *Proceedings the Seventh International Conference on Genetic Algorithms (ICGA97)*, pp.260–267, Morgan Kaufmann.

[89] K. Ohkura, Y. Matsumura and K. Ueda (1998), "Robust Evolution Strategies", *Proceedings of the 2nd Asia-Pacific Conference on Simulated Evolution and Learning(SEAL'98)*, Lecture Note in Computer Science, Vol.1585, pp.10–17, Springer.

[90] K. Ohkura, Y. Matsumura and K. Ueda (2001), "Robust Evolution Strategies", *Applied Intelligence*, Vol.15, No.3, pp.153–169, Kluwer Academic Publishers.

[91] I. Ono and S. Kobayashi (1997), "A Real-coded Genetic Algorithm for Function Optimization Using Unimodal Normal Distribution Crossover", *Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA97)*, pp.260–267, Morgan Kaufmann.

[92] I. Ono, H. Kita and S. Kobayashi (1999), "A Robust Real-Coded Genetic Algorithm using Unimodal Normal Distribution Crossover Augmented by Uniform Crossover: Effects of Self-Adaptation of Crossover Probabilities", *Proceedings of Genetic and Evolutionary Computation Conference (GECCO'99)*, pp.496–503, Morgan Kaufmann.

[93] A. I. Oyman and H.-G. Beyer (2000),"Analysis of the (1,$\lambda$)-ES on the Parabolic Ridge", *Evolutionary Computation*, Vol.8, No.3, pp.249–265.

[94] A. I. Oyman and H.-G. Beyer (2000),"Analysis of the ($\mu/\mu$,$\lambda$)-ES on the Parabolic Ridge", *Evolutionary Computation*, Vol.8, No.3, pp.267–289.

[95] H. Pohlheim (1999), "Visualization of Evolutionary Algorithms- Set of Standard Techniques and Multidimensional Visualization", *Proceedings of Genetic and Evolutionary Computation Conference (GECCO'99)*, pp.533–540, Morgan Kaufmann.

[96] M. Quinn (2000), "Evolving Co-operative Homogeneous Multi-robot Teams", *Proceedings of the IEEE/RSJ Internaional Conference on Intelligent Robots and Systems.*

[97] G. Rudolph (1997), "Local Convergence Rates of Simple Evolutionary Algorithms with Cauchy Mutations", *IEEE Transactions on Evolutionary Computation*, Vol.1, No.4, pp.249–258.

[98] G. Rudolph (1999), "Self-Adaptation and Global Convergence: A Counter-Example", *Proceedings of Congress on Evolutionary Computation (CEC'99)*, pp.646–651.

[99] R. Salomon (1996), "Increasing Adaptivity through Evolution Strategies", *From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, pp.411–420, MIT Press.

[100] R. Salomon (1996), "Performance Degradation of Genetic Algorithms under Coordinate Rotation", *Proceedings of the 5th Annual Conference on Evolutionary Programming*, pp.155–161, MIT Press.

[101] N. Saravanan and D. B. Fogel (1994), "Learning Strategy Parameters in Evolutionary Programming: An Emprical Study", *Proceedings of the 3rd Annual Conference on Evolutionary Programming*, pp.269–280, World Scientific Publishing.

[102] N. Saravanan and D. B. Fogel (1997), "Multi-Operator Evolutionary Programming: A Preliminary Study on Function Optimization", *Proceedings of the 6th Annual Conference on Evolutionary Programming*, Lecture Notes in Computer Science, Vol.1213, pp.215–221, Springer.

[103] J. D. Schaffer and A. Morishima (1987), "An Adaptive Crossover Distribution Mechanism for Genetic Algorithm", *Proceedings of the Second International Conference on Genetic Algorithms (ICGA87)*, pp.36–40, Morgan Kaufmann.

[104] H.-P. Schwefel (1995), *Evolution and Optimum Seeking*, John Wiley & Sons.

[105] H.-P. Schwefel and G. Rudolph(1995), "Contemporary Evolution Strategies", *Proceedings of Artificial life 3*, Lecture Notes in Computer Science, Vol.929, pp.893–907, Springer.

[106] A. C. Slocum, D. C. Downey and R. D. Beer (2000), " Further Experiments in the Evolution of Minimally Cognitive Behavior: From Perceiving Affordances to Selective Attention", *From Animals to Animats 6: Proceedings of the Sixth International Conference on Simulation of Adaptive Behavior*, pp.430–439, MIT Press.

[107] J. E. Smith and T. C. Fogarty (1996), "Adaptively Parameterised Evolutionary Systems: Self Adaptive Recombination and Mutation in a Steady State Genetic Algorithm", *Proceedings of the 4th Conference on Parallel Problem Solving from Nature - PPSN IV*, Lecture Notes in Computer Science, Vol.1141, pp.441–450, Springer.

[108] W. M. Spears (1995),"Adapting Crossover in Evolutionary Algorithms", *Proceedings of the 4th Annual Conference on Evolutionary Programming*, pp. 367–384, MIT Press.

[109] P. Stagge (1998), "Averaging Efficiently in the Presence of Noise", *Proceedings of the 5th Conference on Parallel Problem Solving from Nature - PPSN V*, Lecture Notes in Computer Science, Vol.1498, pp.188–197, Springer.

[110] A. H. Wright (1991), "Genetic Algorithms for Real Parameter Optimization", *Foundation of Genetic Algorithms*, pp.205–218, Morgan Kaufmann.

[111] D. H. Wolpert and W. G. Macready (1996), "No-Free-Lunch Theorems for Search", Technical Report SFI-TR-95-02-010, The Santafe Institute, Santafe, New Mexico, ftp://ftp.santafe.edu/pub/wgm/

[112] D. H. Wolpert and W. G. Macready (1997), "No-free-lunch Theorems for Optimization", *IEEE Transactions on Evolutionary Computation*, Vol.1, No.1, pp.67–82

[113] A. S. Wu, K. A. De Jong, D. S. Burke, J.J. Grefenstette and C. L. Ramsey (1999),"Visual Analysis of Evolutionary Algorithms", *Proceedings of Congress on Evolutionary Computation (CEC'99)*, pp.1419–1425.

[114] J.-M. Yang, J.-T. Horng and C.-Y. Kao, " A Continuous Genetic Algorithm for Global Optimization", *Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA97)*, pp.230–237, Morgan Kaufmann.

[115] X. Yao (1993), "A Review of Evolutionary Artificial Neural Networks", *International Journal of Intelligent Systems* Vol.8, pp.539–567.

[116] X. Yao and Y. Liu (1996), " Fast Evolutionary Programming", *Proceedings of the 5th Annual Conference on Evolutionary Programming*, pp.451–460, MIT Press.

[117] X. Yao and Y. Liu (1997), "Fast Evolution Strategies,", *Proceedings of the 6th Annual Conference on Evolutionary Programming*, Lecture Notes in Computer Science, Vol.1213, pp.151–161, Springer.

[118] X. Yao, G. Lin and Y. Liu (1997), " An Analysis of Evolutionary Algorithms Based on Neighborhood and Step Sizes", *Proceedings of the 6th Annual Conference on Evolutionary Programming*, Lecture Notes in Computer Science, Vol.1213, pp.297–307, Springer.

[119] X. Yao and Y. Liu (1997), "Fast Evolution Strategies", *Control and Cybernetics*, Vol.26, No.3, pp.467–496.

[120] X. Yao and Y. Liu (1998), " Scaling up Evolutionary Programming Algorithms", *Proceedings of the 7th Annual Conference on Evolutionary Programming*, pp.103–112, Springer.

[121] X. Yao, Y. Liu and G. Lin (1999), " Evolutionary Programming Made Faster", *IEEE Transactions on Evolutionary Computation*, Vol.3, No.2, pp.82–102.

[122] X. Yao (1999), "Evolving Artificial Neural Networks," *Proceedings of the IEEE*, Vol.87, No.9, pp.1423–1447.

[123] G. Yin, G. Rudolph and H.-P. Schwefel (1995),"Analyzing the (1,$\lambda$) Evolution Strategy via Stochastic Approximation Methods", *Evolutionary Computation*, Vol.3, No.4, pp.473–489.

# 付 録A　研究成果

## A.1　著書

1. Yoshiyuki Matsumura, Kazuhiro Ohkura and Kanji Ueda, "The Effect of Multi-parent Recombination on Evolution Strategies for Noisy Objective Functions", Book Chapter in Computational Intelligence in Control, Idea Group Publishing, (2002.2)(in printing).

## A.2　投稿論文

1. Kazuhiro Ohkura, Yoshiyuki Matsumura and Kanji Ueda, "Robust Evolution Strategies", Lecture Notes in Computer Science, Vol. 1585, pp10-17, (1999.5).
2. 大倉 和博, 松村 嘉之, 上田 完次, "冗長個体表現を用いた進化戦略による工学的実関数最適化", 計測自動制御学会論文集, Vol.35, No.11, pp1469-1477, (1999.11).
3. Yoshiyuki Matsumura, Kazuhiro Ohkura and Kanji Ueda, "Statistical Characteristics of Evolution Strategies", Lecture Notes in Computer Science, Vol.1917, pp119-128, (2000.9).
4. 大倉 和博, 松村 嘉之, 上田 完次, "遺伝的浮動を利用する進化的プログラミング", システム制御情報学会論文集, Vol.45, No.8, pp409-417, (2001.8).
5. Kazuhiro Ohkura, Yoshiyuki Matsumura and Kanji Ueda, "Robust Evolution Strategies", Applied Intelligence, Vol.15, No.3, pp.153-169, Kluwer Academic Publishers, (2001.11).
6. 松村 嘉之, 大倉 和博, 上田 完次, "Multi-parent Recombination を用いる進化戦略", 計測自動制御学会論文集, (2002.4)(投稿中).

## A.3　国際会議

1. Kazuhiro Ohkura, Yoshiyuki Matsumura and Kanji Ueda, "Robust Evolution Strategies", 2nd Asia Pacific Conference on Simulated Evolution and Learning

(SEAL'98), in Canberra, Australia, (1998.11).

2. Yoshiyuki Matsumura, Kazuhiro Ohkura and Kanji Ueda, "Evolutionary Programming with Non-coding Segments for Real-valued Function Optimization", 1999 IEEE International Conference on Systems, Man, and Cybernetics Conference (SMC'99), in Tokyo, Japan, (1999.10).

3. Yoshiaki Katada, Mikhail Svinin, Yoshiyuki Matsumura, Kazuhiro Ohkura and Kanji Ueda, " An Evolutionary Approach to the Synthesis of Stable Grasp in Multi-Fingered Robot Hand", 5th International Symposium on ARTIFICIAL LIFE AND ROBOTICS (AROB 5th '00), in Oita, Japan, (2000.1).

4. Yoshiyuki Matsumura, Kazuhiro Ohkura and Kanji Ueda, "Statistical Characteristics of Evolution Strategies", International Conference on Parallel Problem Solving from Nature 6 (PPSN6), in Paris, France, (2000.9)

5. Yoshiaki Katada, Mikhail Svinin, Yoshiyuki Matsumura, Kazuhiro Ohkura and Kanji Ueda, " Stable Grasp Planning by Evolutionary Programming", 2000 IEEE International Conference on Industrial Electronics Control and Instrumentation (IECON2000), in Nagoya, Japan, (2000.10).

6. Yoshiyuki Matsumura, Kazuhiro Ohkura and Kanji Ueda, "Evolutionary Dynamics of Evolutionary Programming in Noisy Environment", 2001 IEEE Congress on Evolutionary Computation (CEC01), in Soul, Korea, (2001.5).

7. Yoshiyuki Matsumura, Kazuhiro Ohkura and Kanji Ueda, "The Effect of Multiparent Recombination on Evolution Strategies for Noisy Objective Functions", 2001 International Conference on Computational Intelligence for Modeling, Control and Automation (CIMCA2001), in Las Vegas, USA, (2001.7).

8. Yoshiyuki Matsumura, Kazuhiro Ohkura and Kanji Ueda, "The Effect of Multiparent Recombination on Evolution Strategies", The Fourth IFAC Symposium on Intelligent Autonomous Vehicle (IAV2001), in Sapporo, Japan, (2001.9).

9. Yoshiyuki Matsumura, Kazuhiro Ohkura and Kanji Ueda, "Multi-recombinant Evolution Strategies for Real-Valued Function Optimization", ANNIE '01 Smart Engineering Systems Design, in St. Louis, USA, (2001.11).

10. Yoshiyuki Matsumura, Kazuhiro Ohkura and Kanji Ueda, "$(\mu/\mu,\lambda)$-Evolution Strategies for Noisy Objective Functions", International Workshop on Emergent Synthesis (IWES), in Kobe, Japan, (2002.5).

11. Yoshiyuki Matsumura, Kazuhiro Ohkura and Kanji Ueda, "Advantages of Global Discrete Recombination in ($\mu/\mu$,$\lambda$)-Evolution Strategies", 2002 IEEE World Congress on Computational Intelligence (WCCI), in Hawai, USA, (2002.5).

12. Yoshiyuki Matsumura, Kazuhiro Ohkura and Kanji Ueda, "The Combination of Recombination Operators in ($\mu/\mu$,$\lambda$)-Evolution Strategies", The 6th International Conference on Complex Systems (CS02), in Tokyo, JAPAN, (2002.9)(to appear).

13. Yoshiyuki Matsumura, Kazuhiro Ohkura and Kanji Ueda, "Robust Evolutionary Programming Applied to Artificial Neural Networks", 4th Asia Pasific Conference on Simulated Evolution and Learning (SEAL'02), in Orchid Country Club, Singapore, (2002.11)(to appear).

## A.4 国内講演会

1. 松村 嘉之, 大倉 和博, 上田 完次, "Operon-GA による実関数最適化問題に関する研究", 日本機械学会関西学生会卒業研究発表講演会講演前冊集, pp.237, 神戸大, (1998.3).

2. 松村 嘉之, 大倉 和博, 上田 完次, "非コード化セグメントを持つ個体表現を用いた進化型計算による実関数最適化", 第42回システム制御情報学会研究発表講演会講演論文集, pp.51-52, 京都大, (1998.5).

3. 松村 嘉之, 大倉 和博, 上田 完次, "進化戦略における突然変異のための自律的確率分布の獲得", 1998年度精密工学会秋季大会学術講演会講演論文集, pp.608, 北海道大, (1998.9).

4. 松村 嘉之, 大倉 和博, 上田 完次, "中立型進化戦略への確率的分布選択機構の組込み", 第8回インテリジェント・システム・シンポジウム講演論文集, pp.263-268, 福岡, (1998.10).

5. 松村 嘉之, 大倉 和博, 上田 完次, "進化的プログラミングにおける頑健性向上のための一手法", 1999年度精密工学会春季大会学術講演会講演論文集, pp.519, 東洋大, (1999.3).

6. 松村 嘉之, 片田 喜章, 大倉 和博, ミクハイル シュビニン, 上田 完次, " 進化型計算による多指ハンドの 安定把持形態獲得", 第43回システム制御情報学会研究発表講演会, pp.91-92, 大阪大, (1999.5).

7. 片田喜章, 松村 嘉之, Mikhail Svinin, 大倉 和博, 上田 完次, "進化型計算を用いた把持計画", 日本機械学会年次大会論文集, (I) 巻, pp.673-674, 慶応大, (1999.7).

8. 松村 嘉之, 大倉 和博, 上田 完次, "冗長個体表現を用いた進化戦略の次元スケールと進化ダイナミクスの関係", 1999 年度精密工学会秋季大会学術講演会講演論文集, pp.39, 東北大, (1999.9).

9. 松村 嘉之, 大倉 和博, 上田 完次, "冗長個体表現を用いた進化戦略におけるノイズの及ぼす影響", 第 9 回インテリジェント・システム・シンポジウム講演論文集, pp.285-290, 福井, (1999.10).

10. 松村 嘉之, 大倉 和博, 上田 完次, "冗長個体表現を用いた進化戦略における進化過程の統計的分析", 第 12 回自律分散システム・シンポジウム講演論文集, 沖縄, (2000.1).

11. 松村 嘉之, 大倉 和博, 上田 完次, "進化的プログラミングにおけるノイズ環境下での進化的振舞い", 2000 年度精密工学会春季大会学術講演会講演論文集, 東京電機大, (2000.3).

12. 松村 嘉之, 大倉 和博, 上田 完次, "進化的プログラミングにおける進化ダイナミクスの解析", 第 43 回システム制御情報学会研究発表講演会, 京都, (2000.5).

13. 片田喜章, 松村 嘉之, Mikhail Svinin, 大倉 和博, 上田 完次, "進化型計算を用いた多指ロボットハンドによる安定把持計画", ロボティクス・メカトロニクス (ROBOMEC'00) 講演会, 熊本, (2000.5).

14. 片田喜章, 松村 嘉之, Mikhail Svinin, 大倉 和博, 上田 完次, "進化的プログラミングを用いた多指ロボットハンドによる安定把持計画", 2000 年度日本機械学会年次大会論文集, pp.449-450, 名城大, (2000.8).

15. 片田喜章, 松村 嘉之, Mikhail Svinin, 大倉 和博, 上田 完次, " Robust-EP を用いた多指ロボットハンドの外乱を考慮した安定把持計画", 第 18 回日本ロボット学会学術講演会, Vol.1, pp.53-54, 立命館大, (2000.9).

16. 松村 嘉之, 大倉 和博, 上田 完次," 進化戦略における組換え操作に関する一考察", 2000 年度精密工学会秋季大会学術講演会, pp.459, 名古屋工業大, (2000.10).

17. 松村 嘉之, 大倉 和博, 上田 完次, " Robust-ES を用いたニューラルコントローラの進化ダイナミクス", 第 10 回インテリジェント・システム・シンポジウム講演論文集, pp.223-224, 明治大, (2000.10).

18. 松村 嘉之, 大倉 和博, 上田 完次, "Robust-ESを用いたArtificial Neural Networksの構造進化", 2001年度精密工学会春季大会学術講演会講演論文集, pp.333, 東京都立大, (2001.3).

19. 松村 嘉之, 大倉 和博, 上田 完次, "進化戦略における Multi-parent Recombinationに関する一考察", 第8回MPSシンポジウム講演論文集, pp.107-114, 同志社大, (2001.10).

20. 松村 嘉之, 大倉 和博, 上田 完次, "主成分分析による $(\mu/\mu,\lambda)$-ESの進化ダイナミクス", 2002年度精密工学会春季大会学術講演会講演論文集, pp.158, 東京工業大, (2002.3).

21. 松村 嘉之, 大倉 和博, 上田 完次, "Robust-ESによるContinuous-Time Recurrent Neural Networksの進化的設計", 2002年度精密工学会秋季大会学術講演会講演論文集, 熊本大, (2002.10)(発表予定).

# 謝　辞

研究の場を提供していただき，懇切なる御指導を賜わりました上田完次教授に深く感謝の意を表します．また，博士論文を作成するにあたり貴重な御教示を賜わりました北村新三教授，田浦俊春教授，小島史男教授に謹んで感謝の意を表します．

本研究を進めるにあたり，常に的確な御教示と御指導を賜わりました大倉和博助教授に心より感謝の意を表します．そして，有益な御指導を賜りました鳩野逸生助教授，Mikhail Svinin 助教授，真鍋圭二講師，長坂一郎講師，右田正夫講師に謹んで感謝の意を表します．また，修士の授業以来，統計学に関して御指導を賜わりました垣内逸郎助教授に謹んで感謝の意を表します．

苦楽を共にした旧 **Collective Inteligence Group** 並びに **Robotics & Collective Inteligence Group** の皆さんには様々な面で御協力，激励を頂きました．張明氏，川上賢一郎氏，山田和明氏の先輩諸氏，良き同輩の伍賀正典氏，瀬良香織氏，西川雅史氏，吉川達也氏，弓場孝章氏，また，良き後輩の片田喜章氏，矢島英明氏，安田豊氏，内田潤青氏，河田洋平氏，児島史周氏，保田俊行氏，五十嵐隆之氏，太田将治氏，弘津雄三氏，藤本圭助氏，植田直樹氏，篠原一真氏，谷口友紀氏，山崎謙太氏，鷲崎亮太氏，岩波悟志氏，尾上豪啓氏，高崎真也氏に心よりの感謝を申し上げます．

私が在籍致しました神戸大学大学院自然科学研究科知能システム創成学研究室の歴代諸氏には，様々な面で御協力，激励を頂きました．中村陽一郎氏，沖田淳也氏，黒山和宏氏，藤井信忠氏，岩崎敦氏，竹下豊晃氏の先輩諸氏，良き同輩の生島康教氏，井寄幸平氏，岩本羽生氏，蟹澤美佳氏，木下慎太郎氏，佐藤修一氏，角田真規氏，張萱氏，Zlatan Car 氏，良き後輩の Attila Lengyel 氏，牛尾将蔵氏，河内達磨氏，筒井広城氏，中西大介氏，西野成昭氏，石井洋司氏，小倉武哲氏，津田和城氏，ペ雅樹氏，細井智明氏，三宅史朗氏，吉村悠紀氏，井村修一氏，森卓也氏，小林正明氏，園部隆太氏，また，秘書の河合綾子さん，西田夏菜さんに心から感謝いたします．

最後に，自由気ままに迷惑をかけた家族に心から感謝したいと思います．

2002 年 8 月

松村嘉之．