

A study on search points generating mechanisms in evolutionary computation

大西, 圭

<https://doi.org/10.15017/458554>

出版情報 : Kyushu Institute of Design, 2002, 博士 (芸術工学), 課程博士
バージョン :
権利関係 :

第2章 進化計算研究の現状

本章では、まず進化計算の枠組みを示し、次にこれまでの進化計算研究を概観し、それをふまえて従来の進化計算手法と本論文での取組みとの違いを明らかにする [54]。本論文で提案するあるいは評価する最適化手法は、数値最適化手法である。それらの評価には、ベンチマーク関数を用いる。本論文で用いるベンチマーク関数についてもここで示す。

2.1 進化計算の枠組み

2.1.1 用語説明

ここでは、本論文でしばしば用いられる進化計算研究分野の用語を示す。

- 数値最適化問題 … この問題は、以下に示す n 次元実数ベクトル \mathbf{x}^* を見つけ出す問題として定式化される。これは、関数最適化問題とも呼ばれる。

$$\{\mathbf{x}^* \in \mathbf{R}^n \mid f(\mathbf{x}^*) \leq f(\mathbf{x}), \mathbf{x} \in \mathbf{R}^n, f: \mathbf{R}^n \rightarrow \mathbf{R}\}$$

- 決定変数 … 数値最適化問題において最適化される個々の実パラメータのことである。本論文中では簡単に変数と呼ぶこともある。
- 遺伝子型空間 … 対象とする最適化問題の全解候補集合の進化計算手法内部での表現である。進化計算手法はこの空間内を探索する。このような用途から探索空間とも呼ばれる。この空間を定義し、その要素 (探索点あるいは個体) から表現型空間の要素 (候補解) への写像を決定することをコード化と呼ぶ。
- 表現型空間 … 対象とする最適化問題の全候補解の集合である。これは、解空間あるいは問題空間とも呼ばれる。この空間の要素 (候補解) から遺伝子型空間の要素 (探索点あるいは個体) への写像を決定することをデコード化と呼ぶ。この写像は遺伝子型空間から表現型空間への逆写像である。
- 適応度関数 … 各候補解 (表現型空間内の一点) の最適化問題への適応度を与える仕組みである。候補解へ与えられる適応度は、それに対応する遺伝子型空間内の要素に与えられ、進化計算手法の探索に利用される。適応度は、評価値とも呼ばれる。

- 適応度ランドスケープ … 適応度を持つすべての遺伝子型空間表現のことである。遺伝子型空間上の適応度分布と言うこともできる。
- 探索点 … 遺伝子型空間内の一点のことである。これは個体あるいは染色体とも呼ばれる。
- 遺伝子 … 探索点(個体)を構成する成分のことである。例えば、遺伝子型空間が L ビットのハミング距離空間であるとき、各ビットの値が遺伝子であり、また、遺伝子型空間が N 次元ユークリッド距離空間のとき、各実変数の値が遺伝子である。

2.1.2 進化計算に共通する手続き

進化計算(あるいは進化アルゴリズム)[6]は、生物進化に着想を得た計算方法を総称する呼び名である。その代表的なアルゴリズムには、進化戦略(ES)[66]、遺伝的アルゴリズム(GA)[23]、進化的プログラミング(EP)[17]、がある。進化戦略は数値最適化手法である。進化的プログラミングは最初オートマトン構成手法として提案されたが、現在は数値最適化手法としても利用されている。遺伝的アルゴリズムは、様々な最適化問題に適用可能な概念的な枠組みである。例えば、組合せ最適化問題、数値最適化問題、文字列最適化問題、木構造最適化問題、グラフ構造最適化問題、などの幅広い問題に適用可能である。遺伝的プログラミング [38][24] は、木構造やグラフ構造などの各種構造最適化に特化したものであり、遺伝的アルゴリズムの枠組みに含まれるものである。

進化計算には、生物進化に着想を得ている、という前提がある。しかし、生物学から得られる新たな知識、例えば新しい進化理論 [84] や、純粋な工学的改良により、これまでに様々なアルゴリズムが提案されており、明確にその枠組みを定義することは難しい状況となっている。ただし、進化計算手法と呼ばれるものには、適者の選び方と適者の利用の仕方は別として、ダーウィン進化論の核となる適者生存の考え方が基本にあるように思われる。

ここでは、それら様々なアルゴリズムに共通する基本的な解探索の手続きを述べる。以下に共通する手続きの項目を挙げる。また、図 2.1 に進化計算に共通する基本的な解探索手続きを示す。

1. 複数の探索点を保持する (多点解探索)
2. 現在過去の探索点の情報を利用して新探索点を生成する
3. 保持する探索点あるいは新探索点の生成に利用する探索点の決定に適応度を用いる

ここで注目すべきことは、新たな探索点は、具体的な手続きの違いを無視すれば、基本的には遺伝子型空間(探索空間)内の点から遺伝子型空間(探索空間)内への写像によって生成される、ということである。このとき、新たな探索点が生成される可能性がある探索領域は、そのとき利用される現探索点に依存する。

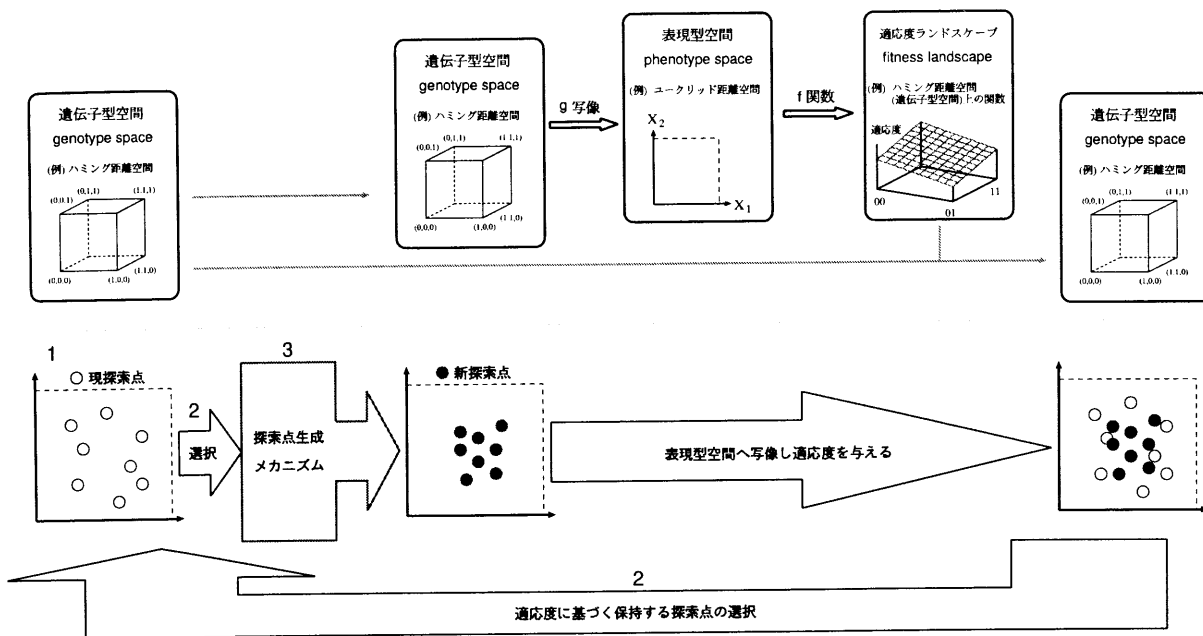


図 2.1: 進化計算に共通する基本的な解探索手続き。

2.1.3 遺伝的アルゴリズムの枠組み

本論文において全ての提案手法の比較対象手法として用いられる遺伝的アルゴリズム (以下 GA) の基本手続きを以下に示す [68]. GA の枠組みは概念的なものである。それは、前節に示した進化計算に共通する手続きと類似する部分を多く持つ。以降の章節で“従来の進化計算手法”と書くときは、ここに示す枠組みを基本的に指すものとする。

1. 初期個体集団の生成

最初の個体 (探索点) 集団をランダムに生成し、それらの適応度を計算する。これを現個体集団 (現個体群) とする。

2. 複製選択

現個体集団から子個体を生成するための親個体を選択し、それらを親個体集団とする。

3. 子の生成

選択された親個体集団に対して交叉・突然変異といった演算を適用して子個体集団を形成し、それらの適応度を適応度関数によって計算する。

4. 生存選択

現個体集団と子個体集団から次の個体集団を形成する個体を選択し、それらを次の個体集団とする。

5. 終了条件を満たすならば手続きを終了する。そうでなければ 2. に戻る。

以上の 2. から 4. までの手続きを 1 つの単位として世代と呼ばれる。現個体集団を構成する個体の数は、個体数あるいは集団サイズと呼ばれる。世代交代モデルの設計とは、複製選択と生存選択の方法を具体的に決定することであり、子の生成法の決定を含まない。子の生成法の設計には、コード化と交叉演算が主に関わる。交叉演算とは、GA における主要演算であり、複数の親個体を用いて子個体を生成する方法である。また、突然変異演算も GA が用いる演算であるが、それは副次的な役割を担う。その役割は、個体集団に斬新な個体を導入することである。

本論文で提案する手法の比較対象として用いられるのは、実数値 GA である。実数値 GA は、遺伝子型空間と表現型空間をともに同じ実数空間 (よってそれらの間の写像は恒等写像) とし、個体を実数ベクトルで表す。また、世代交代モデルとしては、Simple GA (SGA) [18] あるいは Minimal Generation Gap (MGG) [68] を用いる。SGA を用いるとき、エリート k 個体保存戦略を併用することがある。エリート k 個体保存戦略とは、現世代の個体集団の適応度上位 k 個体を次世代の個体集団に強制的にコピーする方法である。また、交叉演算としては、BLX- α 交叉 [14] とシンプレックス交叉 (SPX) [22] を用いる。世代交代モデルとして SGA を用いるとき、親個体に対して交叉演算を適用するかどうかを確率的に決める。この適用する確率のことを交叉率と呼ぶ。また、突然変異演算としては、一様乱数突然変異と正規乱数突然変異を用いる。これらは、個体を構成する遺伝子 (実数パラメータ) に一様乱数あるいは正規乱数で値を付加するものである。これらの適用も確率的に決める。この適用する確率のことを突然変異率と呼ぶ。世代交代モデルである SGA と MGG, 交叉演算の BLX- α と SPX, を以下に説明する。

Simple GA (SGA) [18]

複製選択は、適応度に比例した選択確率で親個体を選択するもの、である。このとき、同一個体の複数回の選択が可能である。生存選択は、子個体集団を無条件で次世代の個体集団とするもの、である。これは、複製選択に適応度を用いるものであり、また、複製選択と生存選択の両方とも個体集団全体に対して適用するものである。

Minimal Generation Gap (MGG) [68]

複製選択は、適応度を用いず現世代の個体集団からランダムに親個体を選択するもの、である。このとき、親個体の中に同一個体はないものとする。生存選択は、親個体集団と子個体集団を合わせた家族の中の最良 1 個体と、家族の中から適応度に比例した確率で選ばれた 1 個体を、次世代の個体集団に残す。このとき、世代交代の対象となるのは、親個体集団であり、SGA とは異なり世代交代が個体集団の部分に限定されている。

BLX- α 交叉 (BLX- α) [14]

2 つの親個体から超直方体を形成し、その超直方体内部に一様乱数によって子個体を生成する (図 2.2(a)). 子個体は以下の手続きによって生成される。以下では n 実変数最適化問題を想定している。

1. 現個体集団から2つの親個体を選ぶ。これらを $\mathbf{x}_1 = (x_{11}, \dots, x_{1n})$, $\mathbf{x}_2 = (x_{21}, \dots, x_{2n})$, とする。
2. 子個体の各変数の値を一様乱数により独立に生成するときの生成範囲を決定する。生成される子個体を $\mathbf{x}_c = (x_{c1}, \dots, x_{cn})$ とする。子個体の第 i 番目の変数 x_{ci} の生成範囲を $[x_{min}^i, x_{max}^i]$ とすると、これらは式 (2.1)(2.2) によって決定される。

$$x_{min}^i = \min(x_{1i}, x_{2i}) - \alpha d_i \quad (2.1)$$

$$x_{max}^i = \max(x_{1i}, x_{2i}) + \alpha d_i \quad (2.2)$$

ただし, $d_i = |x_{1i} - x_{2i}|$, $\min(a, b)$ は a と b の小さい方の値, $\max(a, b)$ は a と b の大きい方の値, α はこの交叉演算のパラメータ, である。

3. 子個体の第 i 番目の変数 x_{ci} の値を, 範囲 $[x_{min}^i, x_{max}^i]$ に一様乱数によって生成する。

BLX- α 交叉のパラメータである α の値は, 本論文でそれが用いられるときに示す。ちなみにこの α の値は, 文献 [14] では 0.5 が用いられており, 文献 [22] では 0.36 が推奨値として導出されている。

シンプレックス交叉 (SPX) [22]

n 実変数最適化問題の場合, $(n + 1)$ 個の親個体によって作られる単体 (simplex) と相似な単体内部に一様乱数によって子個体を生成する (図 2.2(b))。以下に手続きを示す。

1. 現個体集団から $(n + 1)$ 個の親個体を選択する。これらを $\mathbf{p}_0, \dots, \mathbf{p}_n$, とする。
2. 親個体の重心 \mathbf{g} を式 (2.3) により求める。

$$\mathbf{g} = \frac{\sum_{i=0}^n \mathbf{p}_i}{n + 1} \quad (2.3)$$

3. 式 (2.4)(2.5)(2.6) により, $r_k, \mathbf{x}_k, \mathbf{c}_k$ を $k = 0, \dots, n$ について求める。

$$r_k = \{u(0, 1)\}^{\frac{1}{k+1}} \quad (k = 0, 1, \dots, n - 1) \quad (2.4)$$

$$\mathbf{x}_k = \mathbf{g} + \varepsilon(\mathbf{p}_k - \mathbf{g}) \quad (k = 0, 1, \dots, n) \quad (2.5)$$

$$\mathbf{c}_k = \begin{cases} \mathbf{0} & (k = 0) \\ r_{k-1}(\mathbf{x}_{k-1} - \mathbf{x}_k + \mathbf{c}_{k-1}) & (k = 1, \dots, n) \end{cases} \quad (2.6)$$

$u(0, 1)$ は, 区間 $[0, 1]$ 内の一様乱数である。 ε はパラメータであり, 拡張率と呼ばれる。

4. 子個体 \mathbf{c} を式 (2.7) によって得る。

$$\mathbf{c} = \mathbf{x}_n + \mathbf{c}_n \quad (2.7)$$

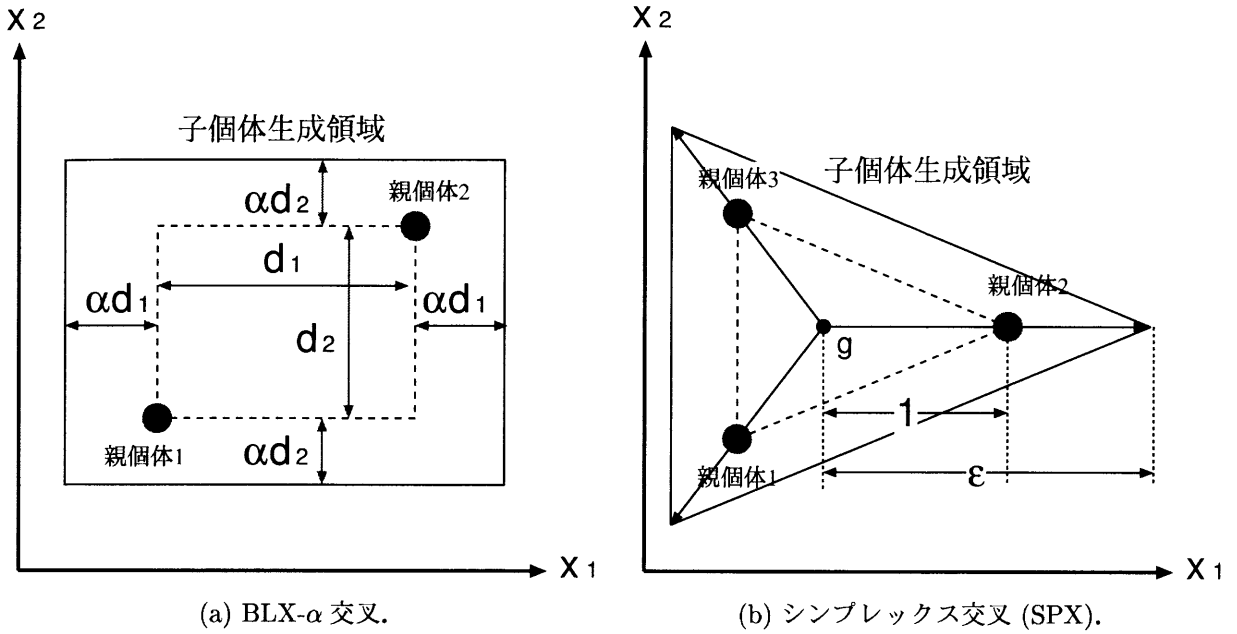


図 2.2: 本論文で用いる実数値 GA の交叉演算.

本論文で SPX を用いるとき、パラメータである拡張率 ε の値として文献 [22] の推奨値である $\sqrt{n+2}$ を用いる.

第 3 章と第 4 章では SGA と BLX- α の組合せを用い、第 5 章と第 7 章では MGG と BLX- α および MGG と SPX の組合せを用いる. MGG と BLX- α および MGG と SPX の組合せにおいては、交叉演算により生成する子個体の数をともに 2 とする.

2.1.4 遺伝的アルゴリズムの設計指針

GA の設計指針として提案されている機能分担仮説 [34][35] とその設計指針に基づく関数最適化問題のための交叉演算の設計指針 [34][35][59] を以下に示す.

機能分担仮説

GA の枠組みは概念的なものであり、世代交代モデルおよび子の生成法として様々なものが設計可能である. 機能分担仮説は、GA の個体集団を確率分布と捉えたとき、世代交代モデルと交叉演算の間には機能分担がある、と主張するものであり、GA 一般に対する設計指針となりうるものである. 世代交代モデルと交叉演算それぞれの機能は以下のようなものである.

- 世代交代モデル

個体集団の分布を徐々に変えて最終的に最適解近傍に導くことがその機能である.

- 交叉演算

個体集団の分布を変えないように子個体を生成することがその機能である。

関数最適化問題の特徴付ける性質としては、(1) 決定変数の連続性、(2) 変数間の依存関係、(3) 座標系のスケール、(4) 局所解の数、(5) 座標系の取り方、が挙げられる。座標系のスケールとは、各座標軸のスケールのことであり、問題によっては座標毎にスケールが異なるものがある。変数間の依存関係がある関数とは、探索空間上のある点において解の改善を行なうためには、複数の決定変数の値を適切に変化させる必要があるものを言う。決定変数についての非線形項を含む関数と言うこともできる。座標系の取り方とは、探索空間の正則線形変換と平行移動によって適応度ランドスケープ構造を変えずに座標変換を行なうことである。

関数最適化問題の性質を考慮した機能分担仮説に基づく交叉演算の設計指針を以下に示す。
関数最適化のための交叉演算の設計指針

- 統計量の遺伝

交叉演算によって生成される子個体集団の分布が、親個体集団分布の統計量である平均ベクトル、分散行列、共分散行列を継承すること、である。

- 多様な解の生成

統計量の遺伝という指針の下で、多様な解を生成可能とすることである。

- ロバスト性の確保

解探索をロバストなものとするために、統計量の遺伝を満たす子個体集団の分布よりもやや広い子個体集団を生成可能とすることである。

関数最適化のための交叉演算の設計指針を満たす交叉演算としては、前述した SPX、単峰性正規分布交叉 (UNDX)[60]、多数の親を用いた単峰性正規分布交叉 (UNDX-m)[33]、がある。また、機能分担仮説が示す機能を満足するような世代交代モデルとしては、前述した MGG がある。

以上に示した実数値 GA の交叉演算設計の際に具体的に考慮すべき点について、これまでに提案されてきた交叉演算はどのように考慮しているかをまとめた文献 [22] に示されている表を表 2.1 に示す。

2.2 探索点生成法とコード化の観点からのこれまでの進化計算研究の類別

ここでは、進化計算手法の探索点生成法とコード化の観点からこれまでの進化計算手法研究を概観する。前節に示した世代交代モデルについても若干示す。本論文で対象とする数値最適化問題のための方法論を中心に取り上げる。

表 2.1: 交叉演算の比較 [22]. BLX- α 交叉の α は, 0.5 として考えられている.

	形質遺伝性	変数間依存性	座標軸方向への依存性	スケールへの依存性	統計量の遺伝	子個体の分布
バイナリ	×	×	?	?	×	?
UNDX	○	○	○	×	○	正規分布
UNDX-m	○	○	○	△	○	正規分布
BLX- α	○	×	×	○	△	一様分布
SPX	○	○	○	○	○	一様分布

進化的演算の改良アプローチ

進化的演算とは, 世代交代モデルと子個体集団を形成する交叉演算や突然変異演算のことであり, それらは一般に確率的演算である.

まず, GA の世代交代モデルとしては, 第 2.1 節で示した SGA [18] と MGG [68] 以外に, Iterated Genetic Search (IGS) [1], Steady State [73], CHC [13], Elitist Recombination [77], Thermodynamical Genetic Algorithms (TDGA) [48], などがある [68]. MGG [68] は, 以上の世代交代モデルよりも個体集団中の適応度分散の大きさの維持 (多様性維持) に優れ, かつ解を改善する能力に優れた方法であることが示されている [68]. これは, 局所解に陥ることを回避し, 最適解近傍に個体集団を集める能力が優れていることを意味する.

進化戦略 (ES) の世代交代モデルとしては, 親個体集団サイズを μ ($\mu > 1$), 子個体集団サイズを λ ($\lambda \geq 1$) とすると, $(\mu + \lambda)$ -ES と (μ, λ) -ES に大別される [5][6]. $(\mu + \lambda)$ -ES と (μ, λ) -ES はともに複製選択としてランダム選択を用いる. $(\mu + \lambda)$ -ES における生存選択は, $(\mu + \lambda)$ 個体集団, つまり親個体と子個体集団を合わせた家族から適応度上位 μ 個の次の親集団を選択するものである. (μ, λ) -ES における生存選択は, λ 個の子個体集団から適応度上位 μ 個の次の親個体集団を選択するものである.

実数値 GA の交叉演算 (crossover) および進化戦略における遺伝子組換え演算 (recombination) には, 第 2.1 節に示した BLX- α [14], SPX [22], UNDX [60], UNDX-m [33], 以外に, Average Crossover [9], Simple Crossover [46], Arithmetical Crossover [46], Real Crossover [83], Linear Crossover [83], Flat Crossover [64], Discrete Recombination [69], Intermediate Recombination [69], Random Intermediate Recombination [69], Global Random Intermediate Recombination [69], Fuzzy Recombination [81], などがある [59]. このうち, 第 2.1 に示した実数値 GA の交叉演算の設計指針を満たす交叉演算は, SPX [22], UNDX [60], UNDX-m [33], だけである.

進化戦略の主探索点生成法は, 突然変異演算である. その突然変異演算として, Correlated Mutation [69][70][5][6] がある. これは, 変数間に相関のある正規乱数を個体 (実数ベクトル) に付加するものである. 正規乱数のパラメータである標準偏差や共分散の値は実数ベクトル (候補解) とともに個体としてコーディングされており, それらのパラメータの値も同様に主に突然変異によって与えられた最適化問題に適応する.

局所探索法とのハイブリッド化アプローチ

進化計算手法に局所探索法を組み合わせる方法である。進化計算手法の一般的な問題として、局所探索能力に劣る、ということがあった。これが背景である。ただし、前節に示した機能分担仮説に基づいて設計されたGAは、個体集団サイズを十分大きくしたとき、局所探索能力にも優れている。

ハイブリッド化の方法はいくつかあり、また実現手法の数も多い。実変数を持つ関数のための局所探索法と進化計算手法とのハイブリッド手法としては、例えば、GAとシンプレックス法とのハイブリッド手法[85][45]、GAと変形パウエル法とのハイブリッド手法[58]、などがある。

局所探索法との組合せとは異なるが、他の方法と組み合わせるという意味で共通する方法としては、解探索の過程で進化計算手法の持つパラメータ(例えば、交叉率、突然変異率)の値をある方法によって変更するものがある[40][89]。また、複数の交叉演算を1回の解探索の中で使い分ける方法がある[86]。

適応度ランドスケープのモデル化アプローチ

ある構造と適応度の与えられた探索点(あるいは候補解)によって適応度ランドスケープをモデル化し、そのモデルに基づいて探索点を生成したり、探索点の適応度を推定するものである。

GAの交叉演算を2次関数による適応度ランドスケープ近似に基づく新探索点生成法とする研究がある[3][72]。これは、親個体から子個体を生成する交叉演算として2次関数を用い、得られた2次関数の極値座標を子個体とするものである。

適応度ランドスケープのモデルを過去の探索点から生成し、そのモデルから新探索点を生成するのではなく、生成された探索点の適応度をそのモデルにより推定し、実際に適応度を計算するかどうかの判断などに使う方法がある。推定に用いるモデルの構造の自由度は一般に高く、遺伝的プログラミングを用いるもの[49]、ニューラルネットワークを用いるもの[57]、既存の探索点からのユークリッド距離及び角度に基づき推定するもの[57][65]、などがある。

適応度ランドスケープのモデルを過去の探索点から生成し、そのモデルから新探索点を生成する方法には、進化戦略と局所探索法のハイブリッド手法に2次関数による近似に基づく探索点生成法を導入する研究がある[41][43][42]。

確率モデルの構築と利用アプローチ

進化計算手法の枠組みにおける子個体集団の生成は従来確率的演算により行なうが、この部分を親個体集団を用いた確率モデルの構築とその構築モデルに基づく子個体集団生成とするアプローチ[62]である。解探索プロセスにおける個体集団の分布は、適応度を基準とした選択によって選ばれた個体であり、それらはある確率分布に従うものと考えられる。その未知の確率分布を推定し、推定により得た確率モデルを探索点生成に利用するアプローチだとみなすことができる。確率モデルの構築は、親個体集団の探索空間内分布を学習するものと

捉えることができ、その学習はあるアルゴリズムにより確率モデルパラメータを決定する手続きである。

このアプローチには、ビット表現あるいはアルファベット表現の遺伝子型空間を持つ最適化問題のための手法 [21][20][61]、実数空間を遺伝子型空間に持つ最適化問題のための手法 [67][71][8][80]、がある。また、アリの群生に着想を得た組合せ最適化手法 [11][10] も、以上の確率モデルの構築と利用アプローチに類似する。

遺伝子型空間構成法

表現型空間とは異なる空間を遺伝子型空間とするアプローチである。数値最適化問題のとき、GAは伝統的にビット空間を用いていた。このとき、コード化(遺伝子型空間と表現型空間の写像)として、バイナリ(2進数)コードとグレイコード [18]がある。これらの遺伝子型空間と表現型空間の間の写像は全単射(1対1対応する写像)である。

コード(染色体)上に遺伝子の値だけでなくその位置(遺伝子座)も表現する Messy GA [19]とよばれる方法がある。つまり、遺伝子の値とその位置のペアが複数配列されたコードである。このとき、遺伝子型空間の大きさは固定ではなく可変となる。ある遺伝子座の値が重複して定義されたり、また、ある遺伝子座の値が不足することが生じる。このような状況に対する対処法が考えられている。

生物の発生プロセスの一部を模倣したコード化がある [87]。これはDNAコーディング法とよばれ、実際のDNAと同様に4つのアルファベットの配列としてコード(染色体)を表現する。模倣される発生プロセスは遺伝子発現(遺伝子からタンパクの生成)プロセスであり、表現型空間への写像をそれと同様な手続きで行なう。このコード化法は、冗長さを持つ。つまり、遺伝子型空間と表現型空間の間の写像関係が多対一である。

ニューラルネットワーク構造最適化問題に対して文法コード化法が提案されている [36] [47]。これは、ニューラルネットワーク構造を生成するための文法によって遺伝子型空間を構成する。文献 [47] に示されている文法コード化の方法を図 2.3 に示す。

進化戦略 [69][70][5][6] は、主探索点生成法として正規乱数付加による突然変異を用いる。この正規乱数のパラメータ(標準偏差や共分散行列)の値は、表現型空間の一点である実数ベクトルとともにコード(染色体)上に表現される。このとき、実数ベクトルは、正規乱数の期待値となる。よって、遺伝子型空間は、実数空間と正規乱数パラメータ空間の直積となる。ただし、各個体(実数ベクトル)の適応度は、正規乱数パラメータの値に影響されない。なぜなら、個体の適応度は、候補解である実数ベクトルの適応度だからである。しかし、新たに生成する個体の適応度は、その元になった正規乱数パラメータ値に影響される。

コード化された文法

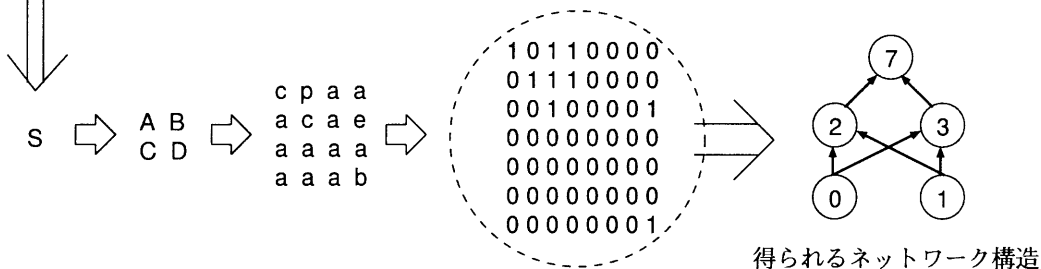
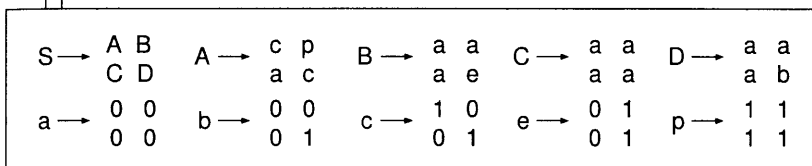
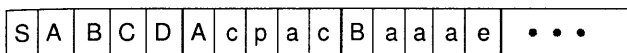


図 2.3: ニューラルネットワーク構造最適化問題における文法コード化法 (文献 [47] より).

2.3 本論文における取組みとの違い

2.3.1 適応度ランドスケープ近似アプローチとの違い

適応度ランドスケープ近似アプローチとは、第3章と第4章における取組みである。従来の進化計算手法と適応度ランドスケープ近似アプローチの間の基本的かつ本質的な違いは、探索点生成に用いる情報にある。従来の進化計算手法の探索点生成法は、探索空間 (遺伝子型空間) 内の点から探索空間内の点への写像法として働く (図 2.4(a))。一方、適応度ランドスケープ近似アプローチの探索点生成法は、適応度ランドスケープ上の点から探索空間内の点への写像法として働く (図 2.4(b))。より具体的には、表現型空間上の適応度分布を、ある構造と適応度の与えられた表現型空間内の点を用いて近似し、近似で得られた構造から表現型空間内の点を生成する。もし、遺伝子型空間と表現型空間が同一ならば、それは本当の適応度ランドスケープの近似となる。

第3章において評価する手法は、適応度ランドスケープ近似に基づく探索点生成法を主探索法の進化計算手法に補助的に導入するものである。

第3章で評価する手法は、第2.2節に示した進化計算手法と局所探索法のハイブリッド手法と、適応度ランドスケープ情報を直接用いる探索点生成法を主探索法の進化計算手法に補助的に導入する、という点で類似する。異なる点は、局所探索手法が適応度ランドスケープをモデル化しないのに対し、評価する手法は適応度ランドスケープをパラメトリック関数による近似によってモデル化すること、である。

第3章で評価する手法と第2.2節に示した交叉演算を2次関数による適応度ランドスケープ近似に基づく探索点生成法とする GA[3][72] との違いは、適応度ランドスケープ近似に基

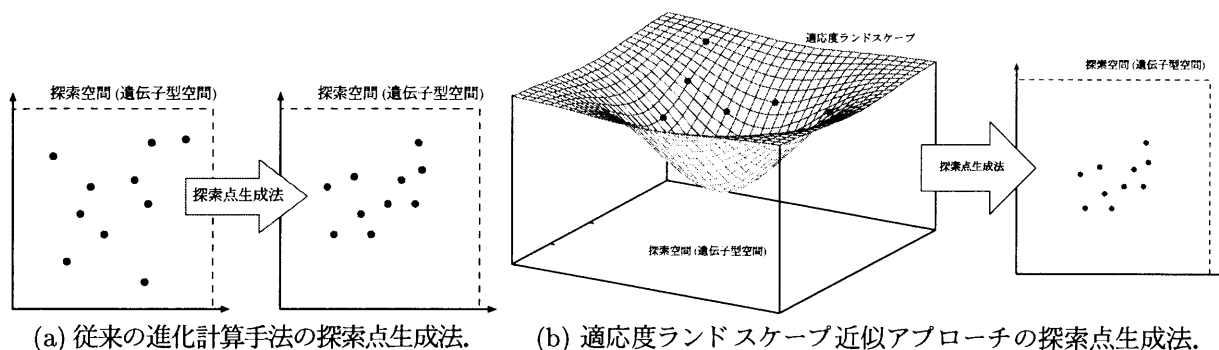


図 2.4: 従来の進化計算手法の探索点生成法と適応度ランドスケープ近似アプローチの探索点生成法の違い。

づく探索点生成法の解探索戦略の中での位置付けにある。この GA は、適応度ランドスケープ近似に基づく新探索点生成法を主解探索戦略 (交叉演算) とする。一方、評価する手法においては、副次的な戦略とする。

第 3 章で評価する手法と第 2.2 節に示した適応度推定にモデルを用いる方法 [49][57] [57][65] との違いは、得られたモデルの利用の仕方にある。適応度推定にモデルを用いる方法は、生成された探索点の適応度をそのモデルにより推定し、実際に適応度を計算するかどうかの判断などに使う。一方、評価する手法は、得られたモデルを探索点生成に用いる。

第 3 章で評価する手法方法と最も類似すると考えられる手法は、第 2.2 節に示した進化戦略と局所探索法のハイブリッド手法に 2 次関数による近似に基づく探索点生成法を導入する手法 [41][43][42] である。このハイブリッド手法における近似に基づく探索点生成法の導入の目的は、多峰性適応度関数に対する解発見の信頼度を上げることである。進化計算手法と局所探索法のハイブリッド手法の実行時の個体集団は、多くの局所解近傍に分布すると予想される。2 次関数による近似に基づく探索点生成の役割は、局所解の近傍間の包絡曲面の近似により新たな有望領域を推定することである。一方、評価する手法では、過去に蓄積した探索点から近似に用いる探索点を選択することで、できるだけ付加的な適応度計算を減らす。過去の適応度の高い探索点を近似に用いるとすれば、結果として上述の方法と同じように局所解の近傍間の包絡曲面の近似を行なうことになる。

第 4 章において提案するアプローチは、適応度ランドスケープ構造学習に基づいて確率モデルを構築し、構築したモデルを探索点生成に用いるものである。このアプローチは、第 2.2 節に示した確率モデルの構築と利用アプローチ [62] と確率モデルを構築し探索点生成に利用するという点で類似する。しかし、確率モデルの構築に用いる探索点が異なる。確率モデルの構築と利用アプローチは、従来の進化計算手法の子個体生成集団生成を行なう演算部分を確率モデルの構築と利用に置き換えるものである。よって、モデル構築には、探索空間内の点を用いる。一方、提案するアプローチは、適応度ランドスケープの点 (適応度を持つ探索

点)を用いて確率モデルを構築する。これは、適応度ランドスケープ近似に基づいて確率モデルを構築するものだと言える。

2.3.2 探索点生成メカニズムの競争進化アプローチとの違い

探索点生成メカニズムの競争進化アプローチとは、第5章における取組みのことである。従来の進化計算手法と探索点生成メカニズムの競争進化アプローチとの間の基本的かつ本質的な違いは、競争進化主体にある。従来の進化計算手法における競争進化主体は、探索点である。一方、探索点生成メカニズムの競争進化アプローチにおける競争進化主体は、複数の探索点を生成可能な仕組みである。具体的には、探索点を生成可能な領域情報とその領域内で探索点を生成する方法とを持つものである。

このアプローチの競争進化主体は、探索点生成領域を構造として持つ点で確率モデルの構築と利用アプローチと類似する。しかし、構造を得る方法が異なる。まず、確率モデルの構築と利用アプローチでは、探索点を競争進化主体とし、競争に勝ち抜いた探索点の分布を構造として学習する(探索点 → 構造 → 探索点 → ...)。この学習は、与えられる学習アルゴリズムによって実行され、進化的手法とは独立である。一方、探索点生成メカニズムの競争進化アプローチは、探索点生成領域を構造として持つ競争進化主体を進化的に強化保持する(構造 → 探索点 → 構造 → ...)。

進化戦略の一実現形態である (μ, λ) -ES[70][5][6]は、探索点生成メカニズムの特性パラメータを解探索と並行して適応させる。これは、探索点を決定する正規乱数が必要とする標準偏差や共分散行列の値を探索点と共に個体にコーディングすることで、探索点を選ばれると同時にその生成メカニズムも強化保持するものである。このとき、個体の一部である探索点は正規乱数が必要とする期待値にもなる。それら正規乱数のパラメータ値は、主に突然変異によって適応する。

以上の (μ, λ) -ESの方法は、進化主体が探索点生成領域を明示的に持つこと、およびそれらが進化的に強化保持される点が提案する最適化の方法と同じである。違いは(1)競争の主体、(2)探索点生成メカニズムの特性を適応させる方法、にある。

まず、 (μ, λ) -ESにおける競争主体は基本的に探索点であり、探索点が競争に勝ち残る結果としてそれに付随する探索点生成メカニズムも生き残ることになる。このとき、探索点と探索点生成メカニズムは1対1の関係にある。一方、提案する最適化の方法における競争主体は、探索点生成メカニズムである。探索点生成メカニズムの適応度を見積もるとき、そのメカニズムにより生成した複数の探索点の適応度を利用するが、このとき探索点と探索点生成メカニズムは多対1の関係にある。 (μ, λ) -ESにおいて1個の個体から複数の探索点を生成すれば、提案する最適化方法と (μ, λ) -ESは、競争主体の観点から類似点を持つと考えられる。

次に、 (μ, λ) -ESにおいては本来の目的である解探索と付随して探索点生成メカニズムの特性の適応が起こるため、その適応は連続的な過程となる。一方、提案する最適化方法は、

(μ, λ) -ES と同様に連続的な適応過程を持つが、第 5 章で示す適応過程の階層性の導入に見られるように断続的な適応過程も持つ。

2.3.3 遺伝子型空間から表現型空間への写像適応アプローチとの違い

遺伝子型空間から表現型空間への写像適応アプローチとは、第 7 章における取組みのことである。このアプローチは、第 5 章の取組みである探索点生成メカニズムの競争進化アプローチに包含されるものである。よって、従来の進化計算手法と遺伝子型空間から表現型空間への写像適応アプローチとの間の本質的な違いは、前節に示したことがらと基本的に同じである。

遺伝子型空間から表現型空間への写像適応アプローチが探索点生成メカニズムの競争進化アプローチと異なる点は、遺伝子型空間の一点となる探索点生成メカニズムによって直接表現型空間内の複数の点を生成するのではなく、遺伝子型空間の一点となるものは写像パラメータ (実際写像元のベクトル) 値生成メカニズムであり、その写像パラメータ値を用いた写像によって表現型空間内の点を生成すること、である。このとき、写像パラメータ空間 (遺伝子型空間) は一般に表現型空間よりも冗長、つまりそれらの間の写像関係が多対一である。実際、その写像関係は、写像法の設計の仕方に依るが、少なくとも伝統的なビットを用いたバイナリコードあるいはグレーコード [18] のように全単射ではない。

第 7 章で実現する手法の中で用いる写像法は、第 6 章で実現する生物の発生機構に着想を得た構造生成手法である。ここで注目した生物の発生機構は、異なる細胞をつくりながらそれらを空間的に正しく配置する仕組み (パターン形成の原理) である。これは、第 2.2 節に示した DNA コーディング法 [87] と生物の発生機構に着想を得ている点で類似する。ただし、注目点が異なる。DNA コーディング法の注目点は、遺伝子発現プロセスである。

第 2.2 節に示したニューラルネットワーク構造の文法コード化 [36] との類似点は、(表現型空間内の一点である) 構造を階層的に分解し、それら階層間にあるルールを用いて結合して最終的な構造をつくるということである。第 7 章で用いる写像法は、各階層の構成を写像プロセスの中で決定するのに対し、文法コード化法では各階層の構成が予め固定されている。また、第 7 章で用いる写像法は、階層間の結合ルールに可変要素を持つが、文法コード化法は固定の結合ルール (文法) を持つ。

2.4 本論文で用いるベンチマーク関数

ベンチマーク関数とは、数値最適化手法の性能評価を行なうために考案される適応度関数である。本論文で示される手法の評価に用いるベンチマーク関数をまとめて表 2.2 に示し、それらの最適解と最適値を表 2.3 に示す。また、図 2.5 と図 2.6 に、各ベンチマーク関数の変数の数 n を 2 としたときのグラフを示す。本論文で示されるすべての手法は、実数値最適化問題の解法であり、それらの手法の評価に用いられる関数はすべて実数値関数である。また、本

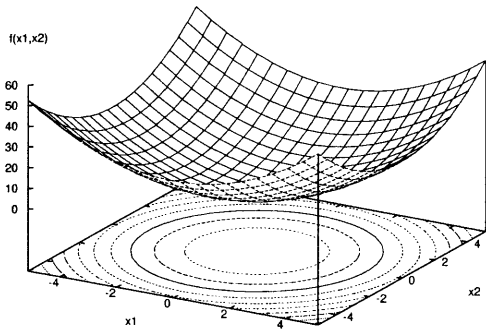
論文で示される数値最適化手法は、それらの関数の最小値を与える n 次元数ベクトルを探索する問題 (関数値最小化問題) に適用される。以降の章節で個々のベンチマーク関数について述べるときは、表 2.2 中のラベル (例えば, F1) を用いる。また、表 2.2 中の探索次元数を表す n の値は、それを実際に評価に用いるときに示す。ここで示すベンチマーク関数以外を評価に用いるときは、そのときに問題の詳細を示す。

表 2.2: 本論文で提案する、あるいは解探索特性を詳しく評価する手法の評価に用いられるベンチマーク関数。

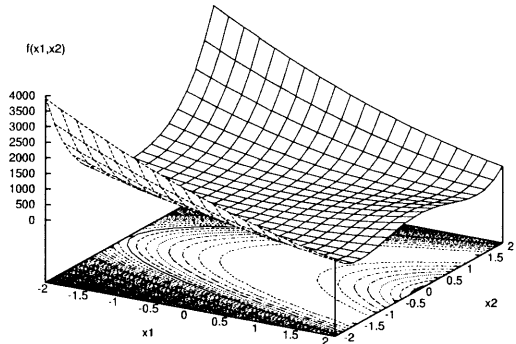
ラベル	ベンチマーク関数	定義域
F1	$\sum_{i=1}^n x_i^2$	$x_i \in [-5.12, 5.12]$
F2	$\sum_{i=2}^n \{100(x_1 - x_i^2)^2 + (x_i - 1)^2\}$	$x_i \in [-2.048, 2.048]$
F3	$\sum_{i=1}^n [x_i]$	$x_i \in [-5.12, 5.12]$
F4	$0.5 + \frac{\sin^2 \sqrt{\sum_{i=1}^n x_i^2 - 0.5}}{1.0 + 0.001 \sum_{i=1}^n x_i^2}$	$x_i \in [-100, 100]$
F5	$(\sum_{i=1}^n x_i^2)^{0.25} [\sin^2 \{50(\sum_{i=1}^n x_i^2)^{0.1}\} + 1.0]$	$x_i \in [-100, 100]$
F6	$10n + \sum_{i=1}^n \{x_i^2 - 10 \cos(2\pi x_i)\}$	$x_i \in [-5.12, 5.12]$
F7	$418.9829n + \sum_{i=1}^n \{-x_i \sin \sqrt{ x_i }\}$	$x_i \in [-500, 500]$
F8	$(x_1 - 5)^2 + \sum_{i=1}^n \{1000(x_i - 5)\}^2$	$x_i \in [-5.12, 5.12]$
F9	$20 + e - 20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp\{\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\}$	$x_i \in [-30, 30]$
F10	$\frac{\sum_{i=1}^n x_i^2}{4000} - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	$x_i \in [-600, 600]$
F11	$-\sum_{i=1}^n \sin(x_i) \sin^{2m}(\frac{ix_i^2}{\pi})$, ただし $m = 10$	$x_i \in [0, \pi]$

表 2.3: 本論文で用いられるベンチマーク関数の最適解と最適値。

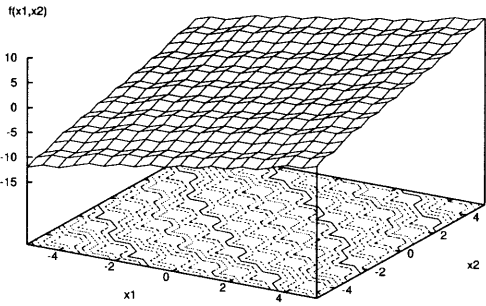
ラベル	最適解	最適値
F1	$\forall i x_i = 0 \ (i = 1, 2, \dots, n)$	0
F2	$\forall i x_i = 1 \ (i = 1, 2, \dots, n)$	0
F3	$\forall i x_i < -5 \ (i = 1, 2, \dots, n)$	$-6n$
F4	$\forall i x_i = 0 \ (i = 1, 2, \dots, n)$	0
F5	$\forall i x_i = 0 \ (i = 1, 2, \dots, n)$	0
F6	$\forall i x_i = 0 \ (i = 1, 2, \dots, n)$	0
F7	$\forall i x_i = 420.9687 \ (i = 1, 2, \dots, n)$	0
F8	$\forall i x_i = 5 \ (i = 1, 2, \dots, n)$	0
F9	$\forall i x_i = 0 \ (i = 1, 2, \dots, n)$	0
F10	$\forall i x_i = 0 \ (i = 1, 2, \dots, n)$	0
F11	変数の数 n に依存	変数の数 n に依存



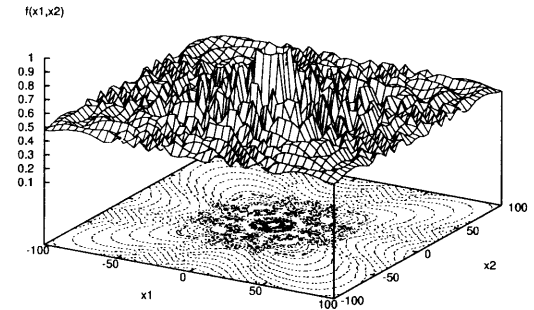
(a) F1.



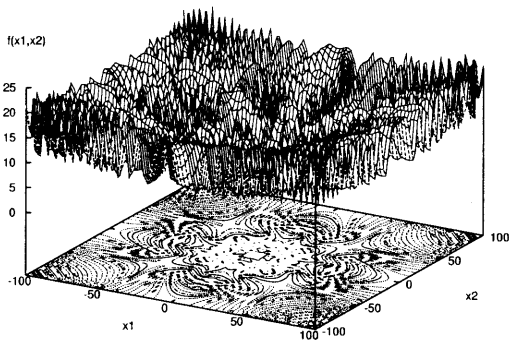
(b) F2.



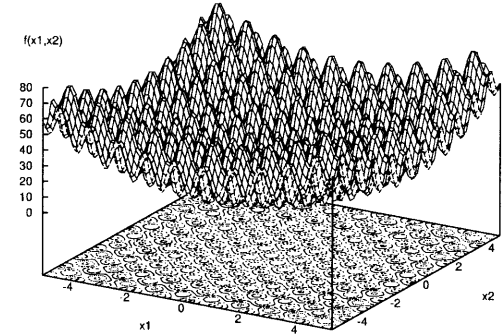
(c) F3.



(d) F4.

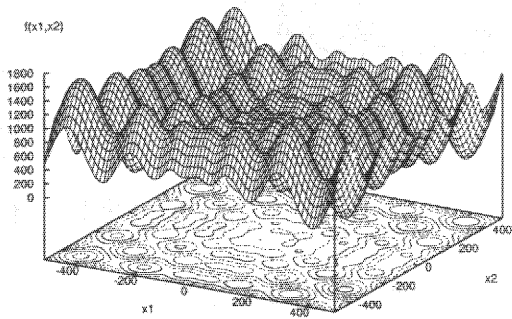


(e) F5.

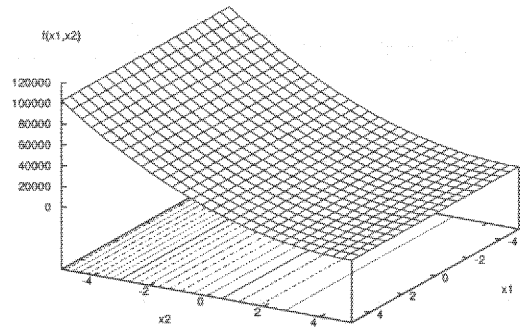


(f) F6.

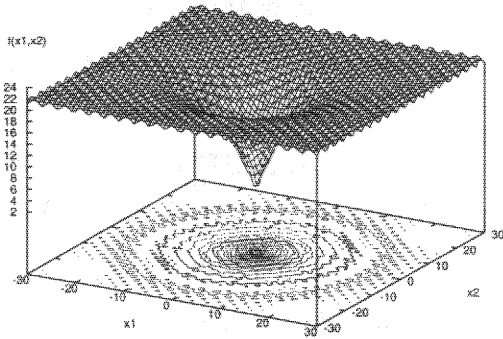
図 2.5: 本論文で用いられるベンチマーク関数 F1 から F6. ただし, 2 変数 ($n = 2$) のもの.



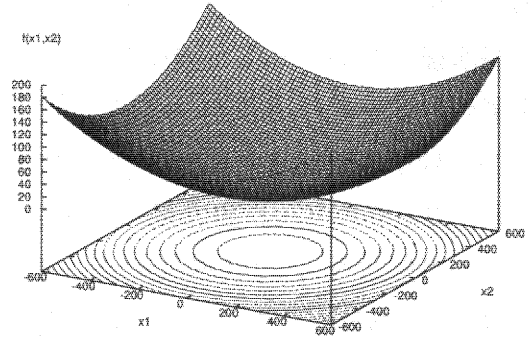
(a) F7.



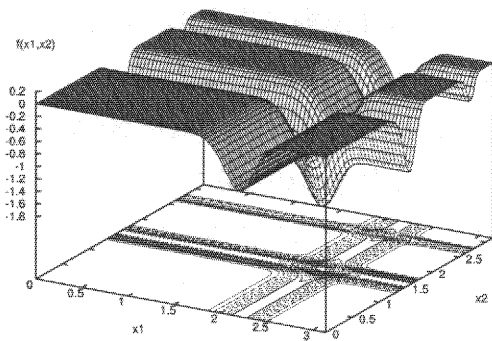
(b) F8.



(c) F9.



(d) F10.



(e) F11.

図 2.6: 本論文で用いられるベンチマーク関数 F7 から F11. ただし, 2 変数 ($n = 2$) のもの.